



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERO EN INFORMÁTICA

Título del proyecto:

Object-oriented development in a JEE5
application of travel booking and management

Alumno: Mikel Otxandorena Sanz

Tutor: Jesus Villadangos Alonso

Pamplona, Julio de 2011

Thanks

First of all thank so much to the Public University of Pamplona for letting me to do the last student year abroad and for teach me a lot of things which have made easier my stay at the company .

I am very grateful for my final project tutor, Jesus Villadangos, because without him it wouldn't have been possible to do my internship here in Paris.

I would like to thank my project manager, Mathieu Bonin for being with me in every second I needed something to continue with work.

I greatly appreciate the three people who interviewed me and let me the opportunity of working at Travelsoft, and tell them that I will not ever forget that interview ever.

I would like to thank also to the Travelsoft Company for offer me a really good contract to start once I will finish my internship.

Thank a lot to all Travelsoft workers, students of the Public University of Pamplona, Erasmus friends, handball team friends, and of course the friends of my city for making these last two years have been wonderful.

I can't forget my parents Eva and Claudio and my brother Imanol because they were the first to cheer me to go my last year abroad and to help me economically.

Finally thank my girlfriend for being so close to me despite the distance and the time we haven't been together. Also I want to tell her that this is for our future.

Index

| | |
|---|-----------|
| Introduction | 7 |
| Travelsoft enterprise | 9 |
| I. Information about the company | 9 |
| History | 9 |
| Activity center | 11 |
| Who are the actors of Travelsoft? | 13 |
| Enterprise organization | 13 |
| II. Work at Travelsoft | 16 |
| Work service | 16 |
| Working method | 16 |
| Tools | 26 |
| Formation | 29 |
| Parts of the project | 29 |
| Reservation | 31 |
| Data structure | 37 |
| What is Java Binding? | 37 |
| First days | 41 |
| Developing work | 42 |
| Version 8.1 and 8.2 | 42 |
| Debug bloc | 43 |
| Reservation step from an xml file | 45 |
| Email sender | 46 |
| Rate details generator | 47 |
| Diagrams of the 8.2 version | 54 |
| Use case diagram | 54 |
| Class diagram | 56 |
| Activity diagram | 57 |
| Deployment diagram | 58 |
| Version 8.2 | 62 |
| Configuration of internal TOs | 63 |
| Exemple1: | 65 |
| Exemple2: | 65 |
| Improve the management of too long fields | 66 |
| Learned: | 67 |
| Send and print rate details | 68 |
| Learned: | 70 |
| Simulator evolution | 71 |
| Learned: | 73 |
| OGONE, evolution of the algorithm to calculate the security key | 74 |
| Example: | 75 |
| Double quote error example: | 77 |

| | |
|--|-----------|
| Sequence diagram: _____ | 79 |
| Learned: _____ | 81 |
| Client Migrations (8.1 → 8.3) _____ | 81 |
| Example: _____ | 82 |
| Learned: _____ | 83 |
| Version 8.4 _____ | 84 |
| Separation of the generic content and specific content _____ | 84 |
| Example: _____ | 86 |
| Learned: _____ | 87 |
| Conclusions _____ | 88 |
| Future lines _____ | 90 |
| References _____ | 93 |

Figure index

| | |
|--|-----------|
| Figure 1: Travelsoft Structure | 11 |
| Figure 2: Activity center at Travelsoft | 12 |
| Figure 3: Organization Travelsoft in Mars 2011..... | 14 |
| Figure 4: Agile development example | 17 |
| Figure 5: Life cycle development project at Travelsoft | 19 |
| Figure 6: Evolution of the 8.3 version | 20 |
| Figure 7: Reservation module evolution curve..... | 23 |
| Figure 8: Conectivity module evolution curve..... | 23 |
| Figure 9: Graphic about where are all tasks | 24 |
| Figure 10: Table about the coefficients | 26 |
| Figure 11: Rate details header for Thalassa..... | 34 |
| Figure 12: Rate details header for Carrefour | 34 |
| Figure 13: First important bloc of rate details page for Thalassa | 34 |
| Figure 14: First important bloc of rate details page for Carrefour voyages..... | 35 |
| Figure 15: Service block on rate details page for Thalassa | 36 |
| Figure 16: Service block on rate details page for Carrefour..... | 36 |
| Figure 17: Total block on rate details page for Thalassa | 37 |
| Figure 18: Total block on rate details page for Carrefour..... | 37 |
| Figure 19: Footer on rate details page for Carrefour..... | 37 |
| Figure 20 : JAXB schema [1]..... | 38 |
| Figure 21 : Unmarshal function..... | 39 |
| Figure 22 : ReservationInfo element example | 40 |
| Figure 23 : Debug block | 43 |
| Figure 24 : Reservation step from a file..... | 46 |
| Figure 25: Email sender interface..... | 46 |
| Figure 26: Functionality in the config file for template generator | 48 |
| Figure 27: Info container element structure | 49 |
| Figure 28 : Gender type for the functionalities on the template generator..... | 49 |
| Figure 29: Rate details template generator interface | 51 |

| | |
|--|-----------|
| Figure 30: Rate details result for the configuration of the figure 22 | 53 |
| Figure 31 : Do tests use case (depth 0) | 55 |
| Figure 32 : Do tests use case (depth 1) | 55 |
| Figure 33 : Class diagram of my work in 8.2 version | 56 |
| Figure 34: Activity diagram of my work in the 8.2 version | 57 |
| Figure 35 : Deployment diagram for the front office server | 59 |
| Figure 36: Deployment diagram for the back office server..... | 60 |
| Figure 37: Deployment diagram for the statistics server..... | 61 |
| Figure 38: Adding columns to the table..... | 64 |
| Figure 39: PackageBooking id sequence..... | 64 |
| Figure 40: Commision example 1 | 65 |
| Figure 41: Commision example 2 | 66 |
| Figure 42: Truncate function..... | 67 |
| Figure 43: Configuration for attach the quote file | 69 |
| Figure 44: Configuration for print quotation | 70 |
| Figure 45: Sequence diagram for Ogone payment..... | 80 |
| Figure 46 : Configuration for static content..... | 86 |

Introduction

I arrived to Paris the last days of August. I was really scared because at first I was going to study computer engineering in English but finally Madame Della Zuana (the person who manages everything for Erasmus students that goes to ESIEE) told me that was impossible. Because of that she offered me studying in French and I decided to come to Paris. One of the things that most motivated me was the structure of the course. The school is oriented to worker life and you can learn things that after you are going to use while you are working, also the internship for 6 months that I have to do on the second semester is really good for my formation and for my CV.

The first days I started with a French course and I learned a lot of interesting things but my French level was really bad. So then I decided to start preparing my CV and covering letters to start sending them to the different companies to have a chance to get an internship. I started sending only to the companies that I thought will be good for me, with the style of work that I was thinking that I was going to enjoy. Finally two companies asked me to have interviews and between them there was really a big difference (situation, money, type of work). When I finished the interviews I waited really nervous but both called me to work with them, so I decided to go to Travelsoft.

So now I am doing my internship in Travelsoft Paris. Travelsoft is a service company specializing in the implementation Applications of selling travels online. Its clients are travel agencies, Tour Operators and airlines that want to sell their products on the internet.

The purpose of the report is to present the company and the work that I have done during my internship. I am going to stay six months in Travelsoft and now I am going to explain my first three months that I was part of a reservation team as developer engineer.

I can say that I have worked for a new project that is going to be very useful for the enterprise and after that I think that I will continue on the reservation part but working in different projects. Here after the company presentation I will resume the entire project that I did step by step. To finish the report a small tutorial will be added to explain how can be used the part where I have worked.

Travelsoft enterprise

I. Information about the company

History

Travelsoft is, as its name suggests, a computer service company specializing in the travel industry. It was created in 2000 by Marko Vujasinovic, Christian Sabbagh and Pablo Carrington with the support of Didier Pineau-Valencia. In the last eight years, it has developed unique expertise in the field of new technologies in the travel industry. Young and dynamic, the company was opened for consulting operations and project management since September 2003.

After 2005, Travelsoft launched a distribution platform of new generation for use by producers and distributors of travel.

In 2007, Travelsoft has expanded its business by creating a project in the field of online recruitment.

Travelsoft has almost 60 employers from more than six nationalities different. They have developed cutting edge expertise in intermediation solutions dedicated to the travel industry. The inter-relationship between providers and distributors is at the heart of the know-how. The knowledge of specialist problems and challenges has helped them develop two dedicated solutions to meet your specialist needs: Tour Cameleo and Air Cameleo.

The Tour Cameleo solution was designed to enable consultation and reservation online for all channels of a distributor, and in particular channels aimed at

travelers, those used by travel agents, but also indirect distribution channels. The management of all distribution channels, users, their roles as well as the travel policy of each channel is carried out in the Tour Cameleo administration tool.

Travelsoft has capitalized on its experience in flight search engine projects to create a very innovative airline booking engine and this part is called Air Cameleo.

Travelsoft advises leading companies in the travel industry sector, as well as - in business travel - major companies, and helps them implement changes in organization, processes and information systems.

The travel industry is currently facing various challenges: automation of the distribution of products through multiple channels, greater market flexibility and reactivity, dynamic packaging, changing rates, new approaches to customer relations, globalization, and security management.

Travelsoft offers a wide range of on demand services covering each stage in the life of your transformation projects, from design to full implementation. On the figure number 1 that we have on the page 11 is illustrated the work that Travelsoft does.

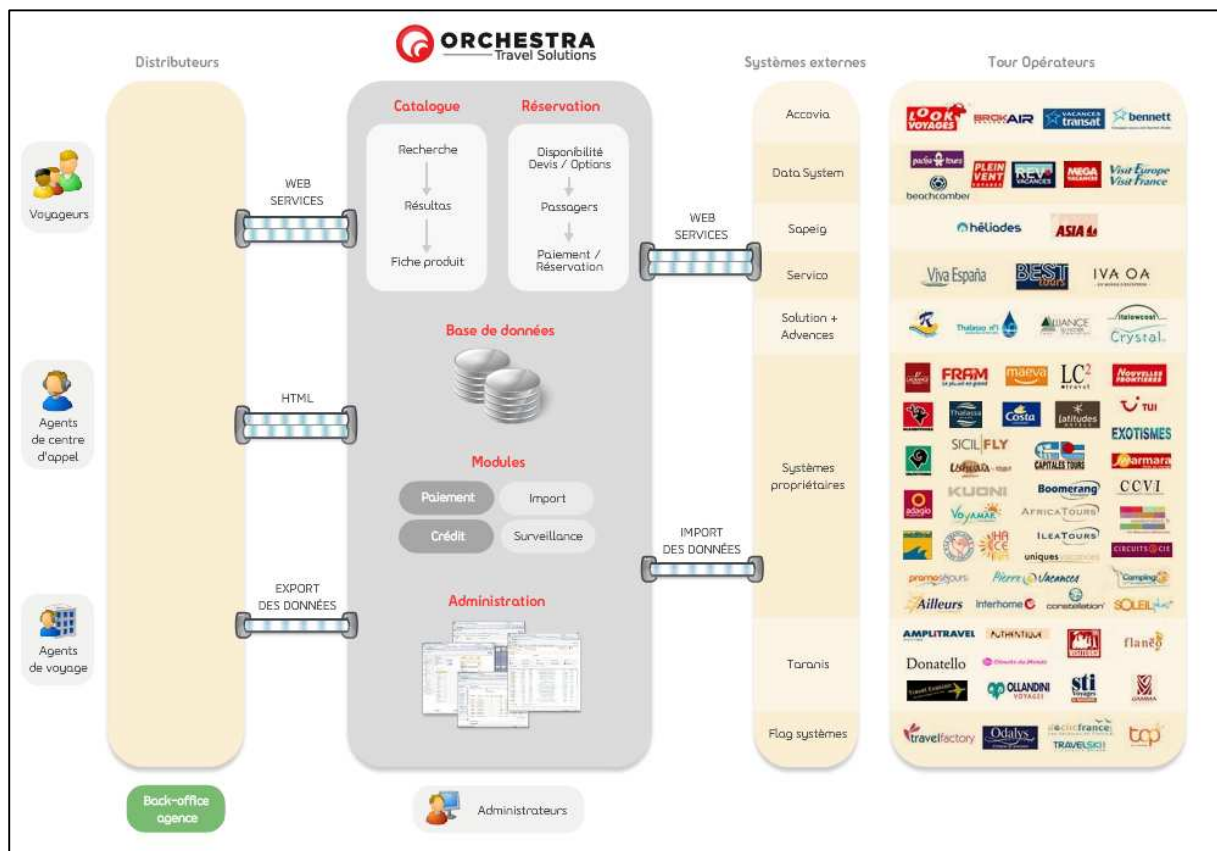


Figure 1: Travelsoft Structure

Activity center

Travelsoft uses modular platform to offer several types of services to its customers. Among those services we can find:

- Achieving IT Project type website on behalf of clients from the travel industry whether booking applications and purchase online B2B, B2C, B2D B2E or, intranet portals/extranet or electronic catalogs.

- Support and maintenance for the sites those are online. Travelsoft offers accommodation with the support and maintenance of these sites ensuring their availability (> 99.5%).
- Support for project management: business consulting with businesses needing to make significant technological choice or making calls for tender.

We can add to these activities, developments and internal development introduction of new platforms or reusable components maintenance them if they already exist. On the image number 2 that is on the page 12 there is a small graph in French who explains the entire activity center that Travelsoft has.

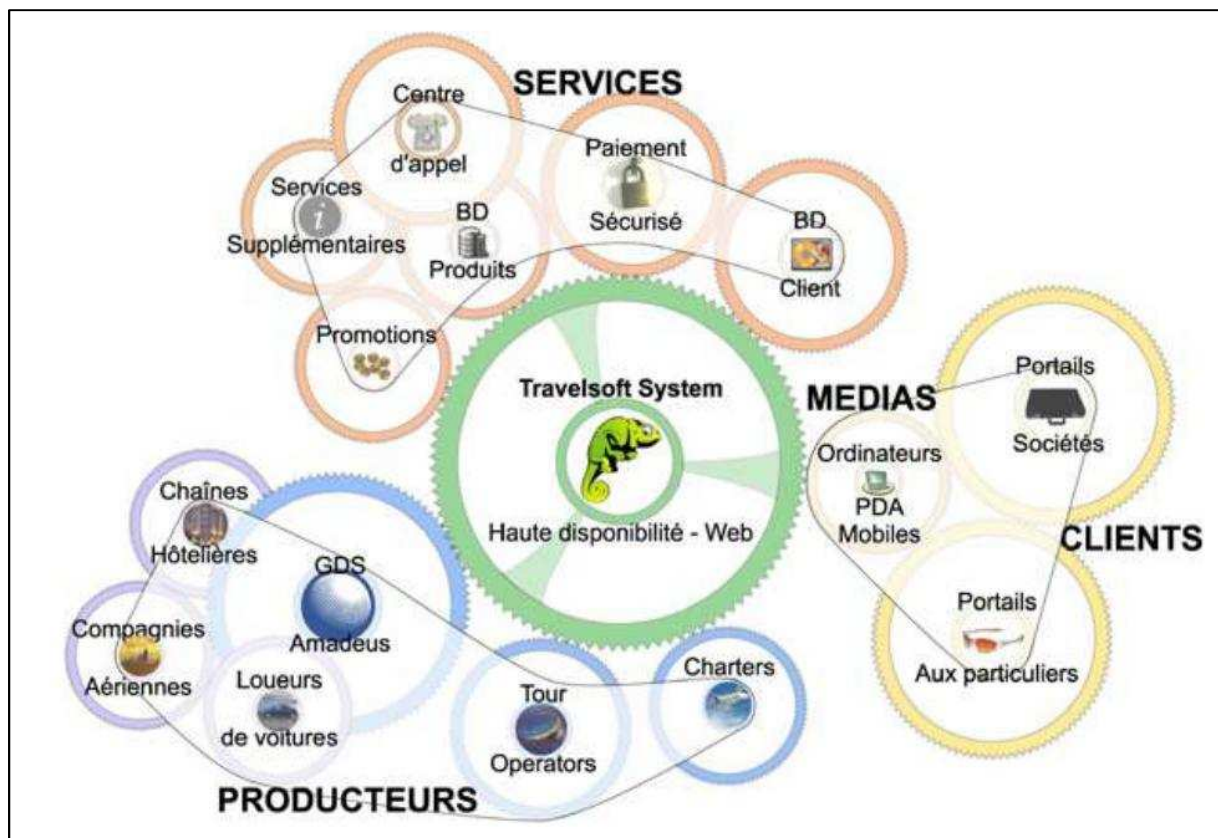


Figure 2: Activity center at Travelsoft

Who are the actors of Travelsoft?

Travelsoft at the activity center works with travel producers and sellers. Among them, we can find:

- The suppliers: airlines, airports, hotel companies, business car rental, insurance, etc...
- The Tour Operators (TO): are companies that are specialized in establishment of tourism packaged products. A “package” is an offer multi-product involved in a holiday. Those Tour Operators can take care of airfare, accommodation, catering, insurance, rental car or equipment, activities, leisure packages, etc... There are a lot of them like: FRAM, Thalasseo, Sti Voyages, adagio, Heliades, etc...
- Tour Distributors (TD) are traders; they distribute the products under from a webpage that all the people can visit. A TD can sell its own products and those of other TO too as for example FNAC, Thalasseo. In the other side we can have TD without their own products like Opodo, Carrefour, CWL, etc, and Travelsoft connect those TDs with some of the TOs that we have in our data base.

Enterprise organization

Travelsoft is a company with almost 60 employees. This is a small dynamic company which one of the strengths is the organization. Here you can find the structure of the company explained on the figure number 3 that is on the page 14 in a small chart:

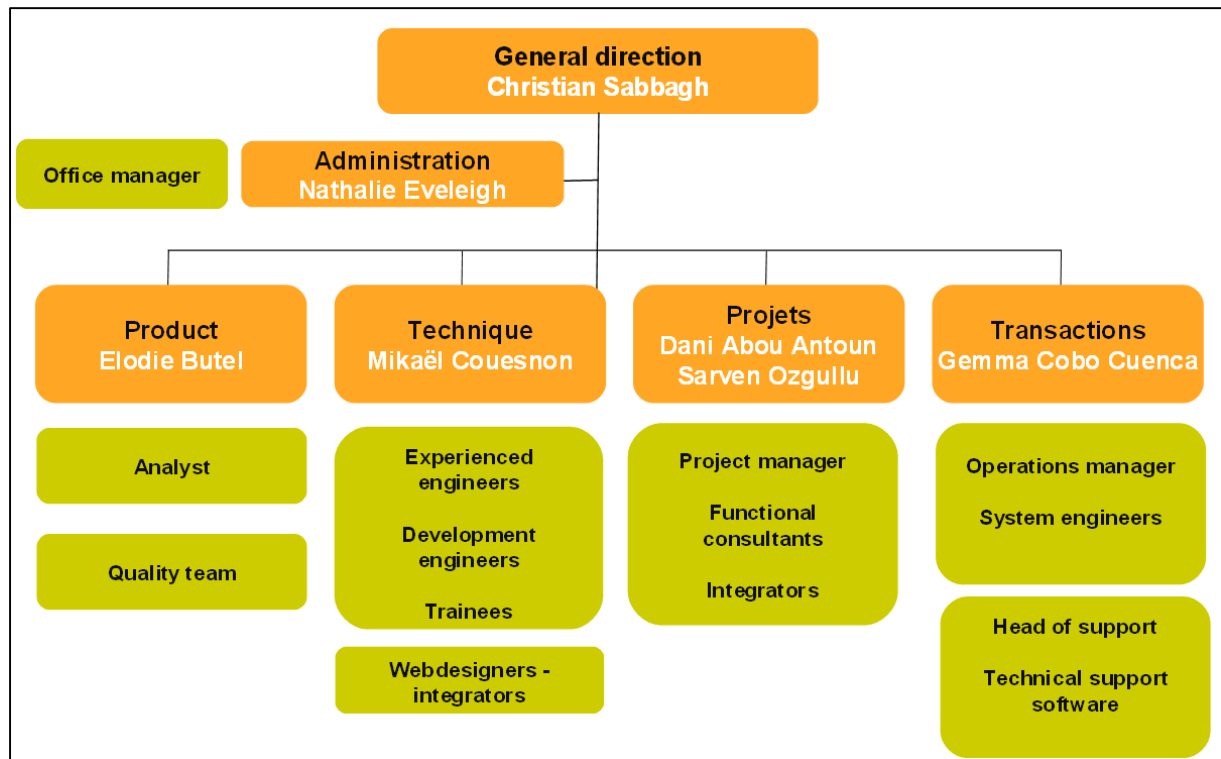


Figure 3: Organization Travelsoft in Mars 2011

Christian Sabbagh is the responsible of all the company and he normally work hard to have good positioning in the market, acquire new customers and to have a good consistency at the enterprise among others.

The principal task of the office manager is to do good reports about all the information that could be interesting for the company, make future previsions and something about the recruitment.

Elodie, the responsible of the product that Travelsoft sells, works to be the company innovative, she tries to improve the different connections with new customers and see things on product quality.

Mikaël is the manager of all the developers. He is the principal person on the agile software element; anything goes on to develop without his consent. He tries to have the entire project scalable and to anticipate the future.

Dani and Sarven are projects directors and at the same time project managers. If in any project happens any problem, they are the responsible to communicate to the correct person to fix as soon as possible.

To finish with the entire tree, Gemma is the responsible of all the operations. She is the head of the support and she works a lot with the IT group to have all the projects working correctly.

II. Work at Travelsoft

Work service

Travelsoft is one of the first companies that use the Java 2 Platform, Enterprise Edition [1] or Java EE in 2001 and all projects use this technology.

Over the years, many components have been developed. They are in a constant evolution trying to adapt that use in the different projects. The organization in modules allows reuse old works and to share development efforts. In order to minimize the code specific to each project exists a global

Working method

After 2005 the company decided to start with a new management method called eXtreme Programming. Their work was OK but they made some investigations and they decided this year to change the method to something similar to SCRUM [2], of course, always continuing the structure of an agile software development. From now on, I am going to use the word SCRUM to refer to the work methodology because I think that is the closer agile methodology to that used at Travelsoft. If there is any doubt, the figure number 4 on the page 17 will show you how goes on an agile methodology.

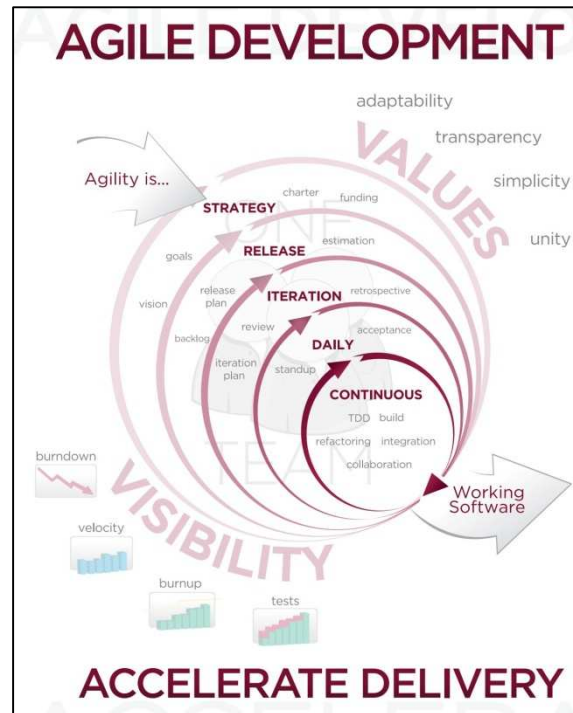


Figure 4: Agile development example

This was a hard decision but finally after months of investigation they thought that SCRUM was better because it doesn't have as much fix points as eXtreme programming. It means that with SCRUM you are freer to decide things than eXtreme programming.

However the method that was implanted a month ago is not exactly SCRUM, it is a method that Travelsoft made based on it. The new process is divided in different stages. On the first stage the director, the product responsible and the development responsible decide which are the highest priority tasks. All of them will be saved on Jira that is the place where every tasks to do are saved and where the people of the company write tasks to do on the future if they have any good idea. With this one I want to explain that all the tasks to do come from Jira, or because any worker thought that the task was interesting to develop and they wrote there or also, because any responsible decided that is interesting to develop anything for the next version and they wrote there too.

After the first meeting where the most important tasks are defined, there is another meeting with the entire module responsible. There each one explains which tasks are important for them and which ones no. After this meeting, it is normal to change the importance of every task because the module responsible has a good knowledge of his own module and nobody knows better than him what is the most important for the module. Of course always they can be Exceptions if the client has any requirement about any module.

Below another meeting is done with the same actors of the first one. Then they confirm all the tasks to do for the next version and they send them to the corresponding module. Once the module responsible has received the tasks he or she decides who is going to be the developer for that task.

The process has 25 days of developing duration. The deliveries are estimated to be done on the last days of each process. The problem is that if we don't finish the tasks in the estimated days, we will have problems there and that is not advisable.

One of the most important parts of the process is the continuous court meetings. Three times a week for 15 minutes all the developers we got together to speak about the tasks that we are developing in that moment and the problems that we can have.

To understand better how works the company; I have attached a graph that is the figure number 5 on the page 19 with the different steps that has a task once someone has started to develop it.

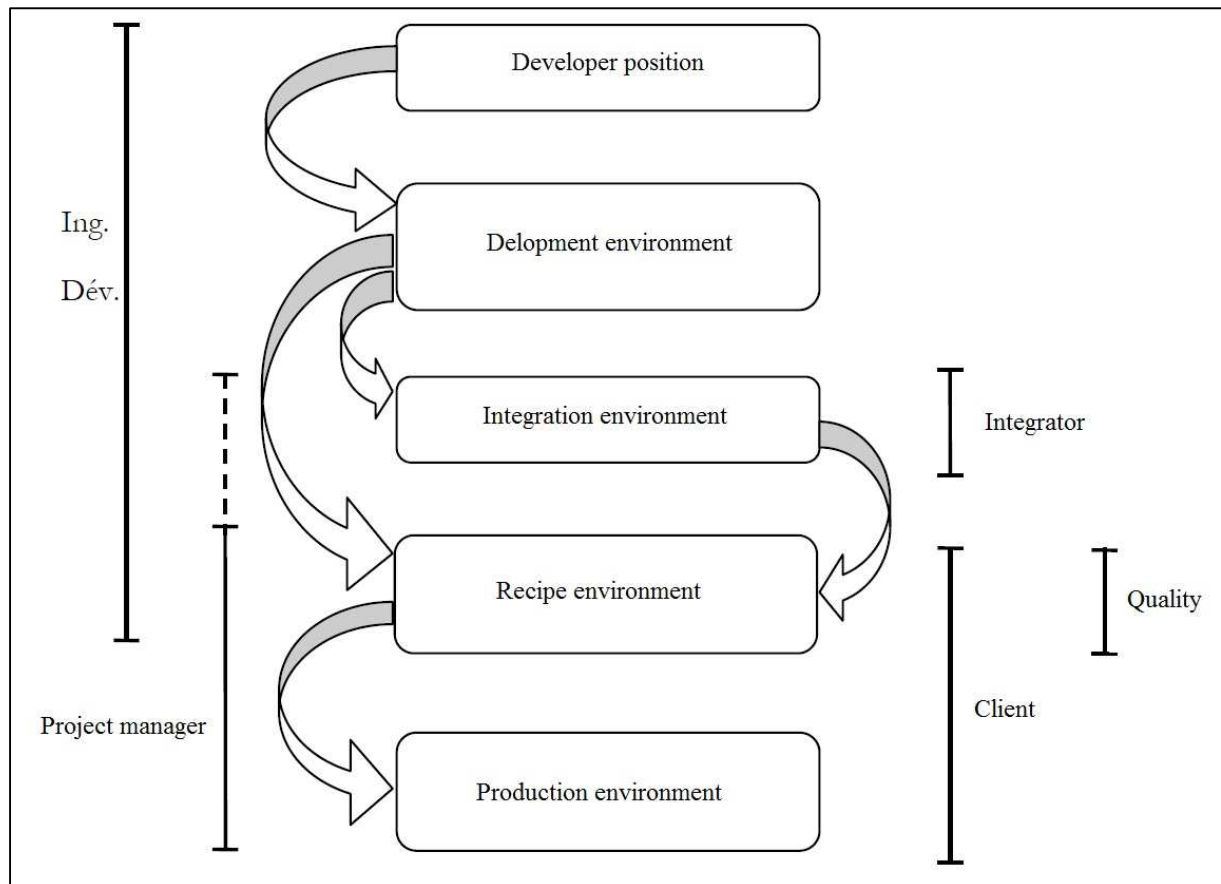


Figure 5: Life cycle development project at Travelsoft

As I think that this is an important part of the report I am going to explain how work the first and second version of this new methodology at Travelsoft. I know too that reading a lot about any methodology without any image which can help us to understand better its hard and because of that I am going to base my next explanation on the following graphics.

On the graphic of the figure 6 at the page 20 we can see a lot of information. On the y-axis, we have the number of days of developing work that there are to develop and on the x-axis there are all the days of the version. If we mix both we can see easily an evolution over the time.

Heeding the lines we can see that we have four. The **straight line** is the perfect evolution of the project but it's impossible to have this because normally the tasks are

longer than 1 developing day and they have always things to test. It's important to say that the line doesn't drop until the days of the task are validated. The **blue curve** is the evolution of the 8.3 version. The **red curve** is the number of days of delay we have comparing with the perfect line. To finish with the curves there is a **dashed curve** that explains the evolution of the 8.2 version over the time.

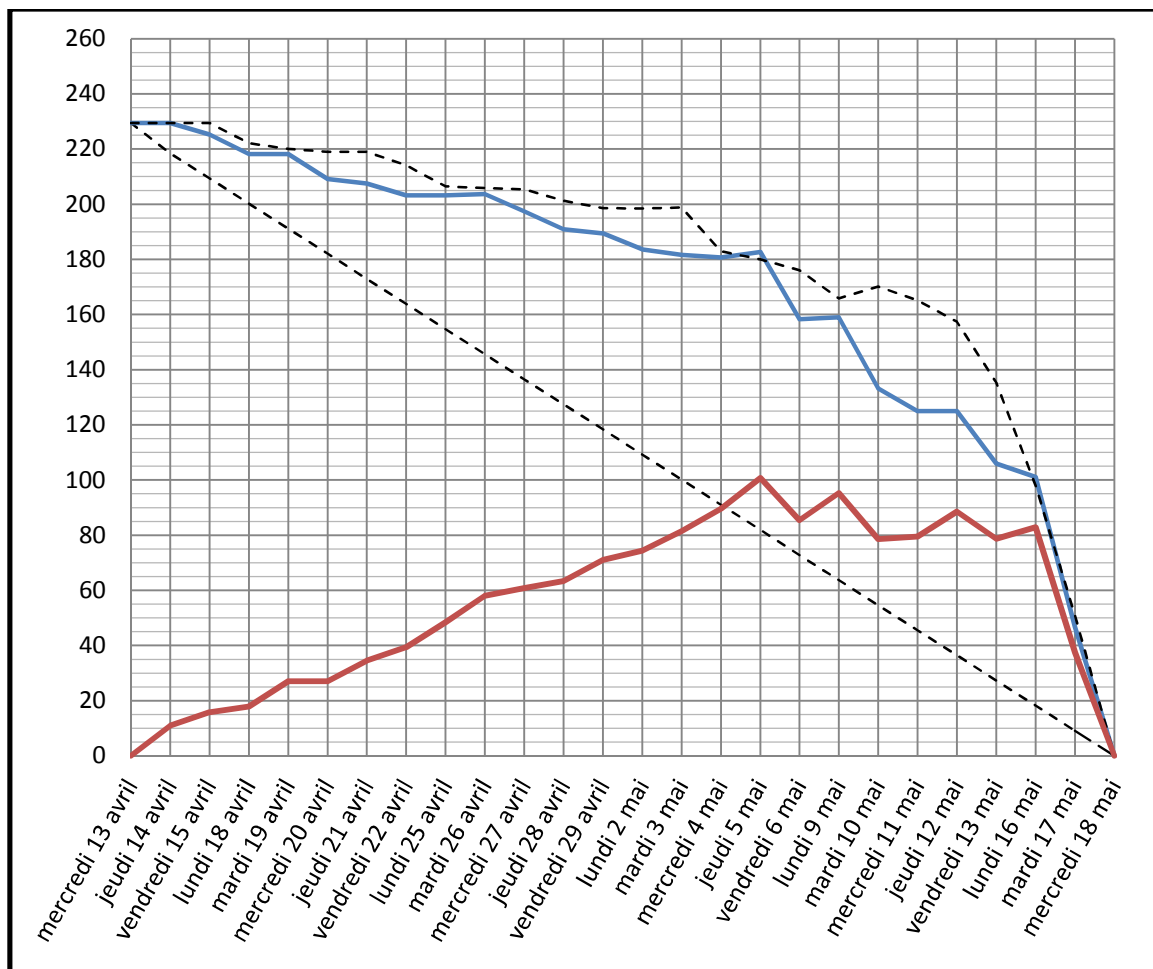


Figure 6: Evolution of the 8.3 version

I think that is interesting to evaluate the results of each version for make better the methodology. Because of that I am going to tell you my impressions about the curves.

About the straight line there isn't so much to tell; only that it's almost impossible to have that evolution. This is because as I tell before, each task normally has between three and four developing days (average) and the developed days doesn't go down until each task is developed, tested by the quality team and validate by them too. So, imagine that I am starting with a 3 days task and I finish the develop of the task on time; I will have "lost" three days on the representation of the curve on the graphic because they still not be validated.

The red curve as I explained before is the curve of the delayed developing days. I don't like to use this name for the curve because is not true that the days are delayed but I think that is enough to understand what means. The time evolution of the curve it's normal.

At first when the developers start developing, the quality time doesn't have much work unless there are one day or half day developing tasks. Of course normally there are but it isn't normal to start with them. But once the days go on some of the tasks will be being tested by the quality team and more and more. This is because the line goes up and finally straight down. So we can resume this telling that the first days there aren't tasks to test, on the half part of the version there are much more and there isn't enough people to validate them quickly and because of that the line goes up. On the last days, if the developing work is well done, there won't be tasks arriving to the quality team because all of them are already done so the only think that happens there is to unstuck work to validate. Therefore the last three or five days the line goes straight down.

Good solution for improve this methodology is to hire more people for the quality team to finish validating the tasks faster and to don't have a lot of work on the last days. But this isn't really normal because the first days of each version we won't have work for all of them, so as this solution isn't normal, the quality team will continue programming unit test the first days of each version and they will continue testing and validating tasks.

The last two curves are about all the evolution in whole version as I explained before. In my opinion the version 8.3 was better done (blue curve) because starting almost with the same developing days, the tasks started being validated before. Also there is a thing that I want to explain and is that on the 4th may there were some days added. This was because the team where were added (the reservation team) was working fast and we had “won” days. It means that until the 4th may we put on test more days than the estimated and we added two more tasks to the version.

As I explained before when I was speaking about the straight line, the last three or five days about each version are days where the quality team has a lot of work and almost a third part of the days are there validated.

One of the reasons is the number of people on the quality team but this isn't the only one. For example on the figure number 7 on the page 23 we can see that little by little the curve goes down in spite of the two added tasks on the 4th may and 6th may. But the connectivity team, as we can see on the image number 8 that is on the on the page 23 they have bigger tasks and this is impossible to develop faster. That is well expressed on the delayed line, as we can see that line goes directly up while each person of the team is developing the biggest tasks but suddenly 12th May one of them is validated and the 17th three.

I remember that one person of that team; he works the entire version in an 18 days task, so it's mathematically impossible to add before 18 days on the test column. Also it's important to explain the last days of connectivity team because in spite of having big works isn't normal to validate 30 days the last two days. This is because as they work with the clients and sometimes they have to wait them to finish their part or answer some questions. Indeed, on this version that happened to three tasks, two of them about the same connector.

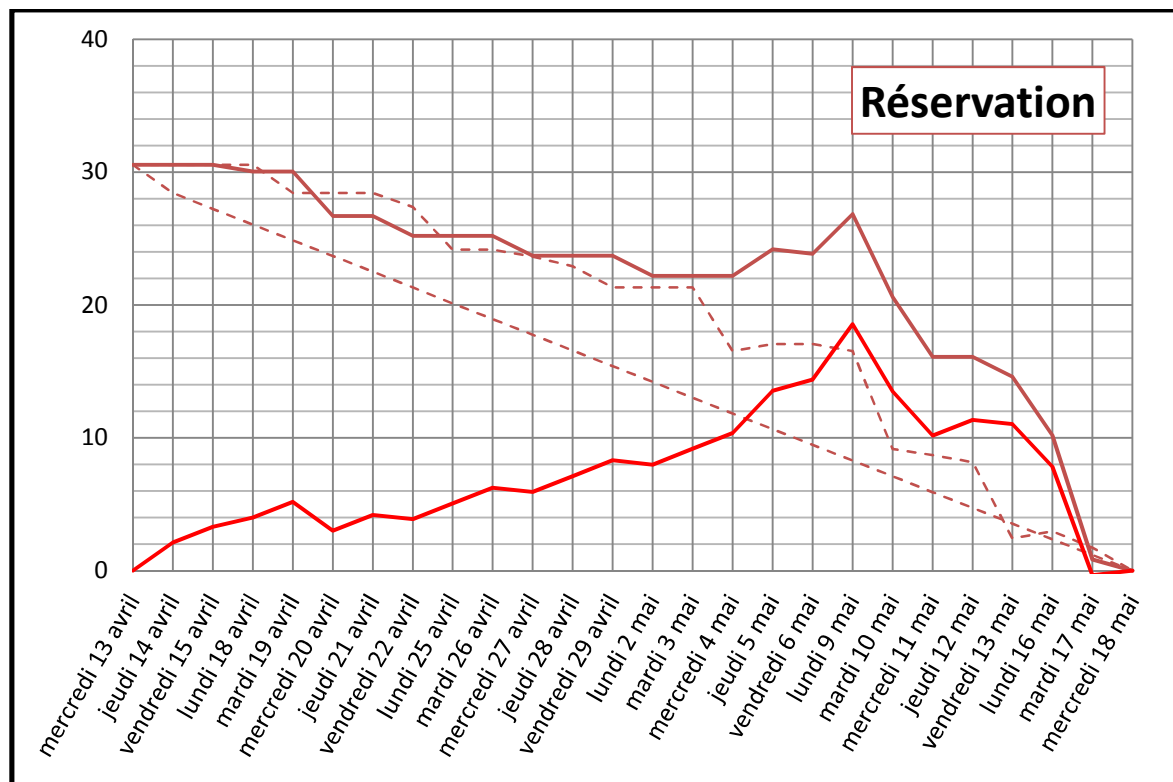


Figure 7: Reservation module evolution curve

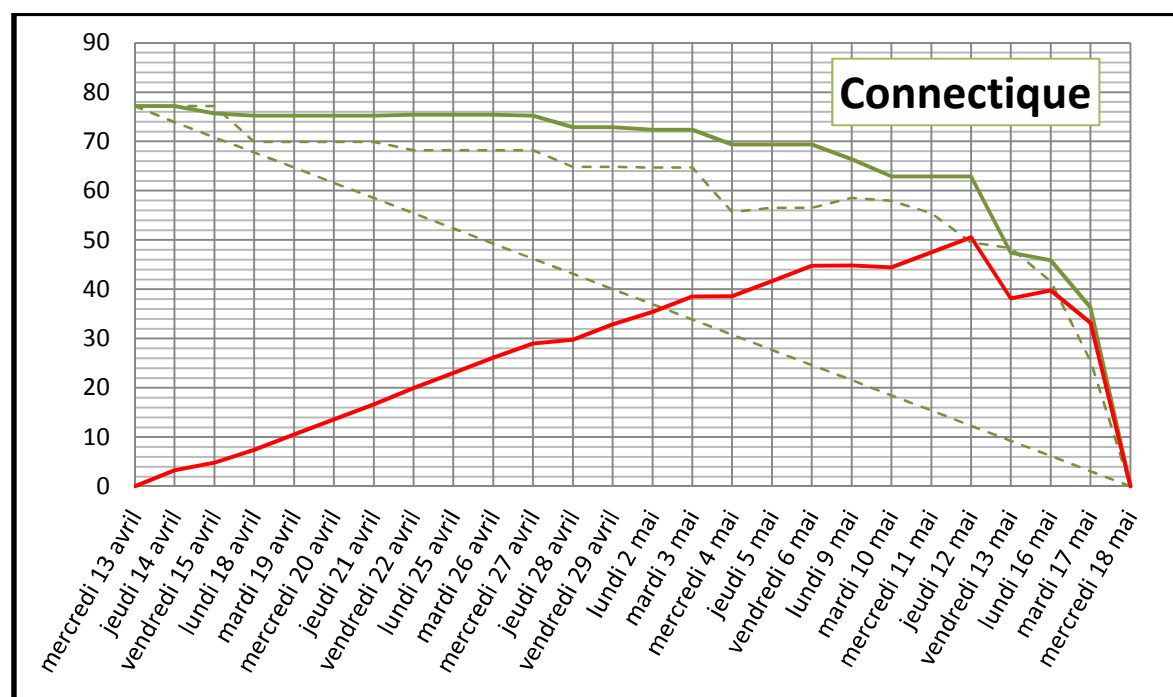


Figure 8: Conectivity module evolution curve

To finish the explanation about the methodology I am going to add the next graphic that is the graphic that we can see on the figure 9 on the page 24, because I think that it's interesting to finish understanding any doubt that anyone can have after the explanation about the evolution curves.

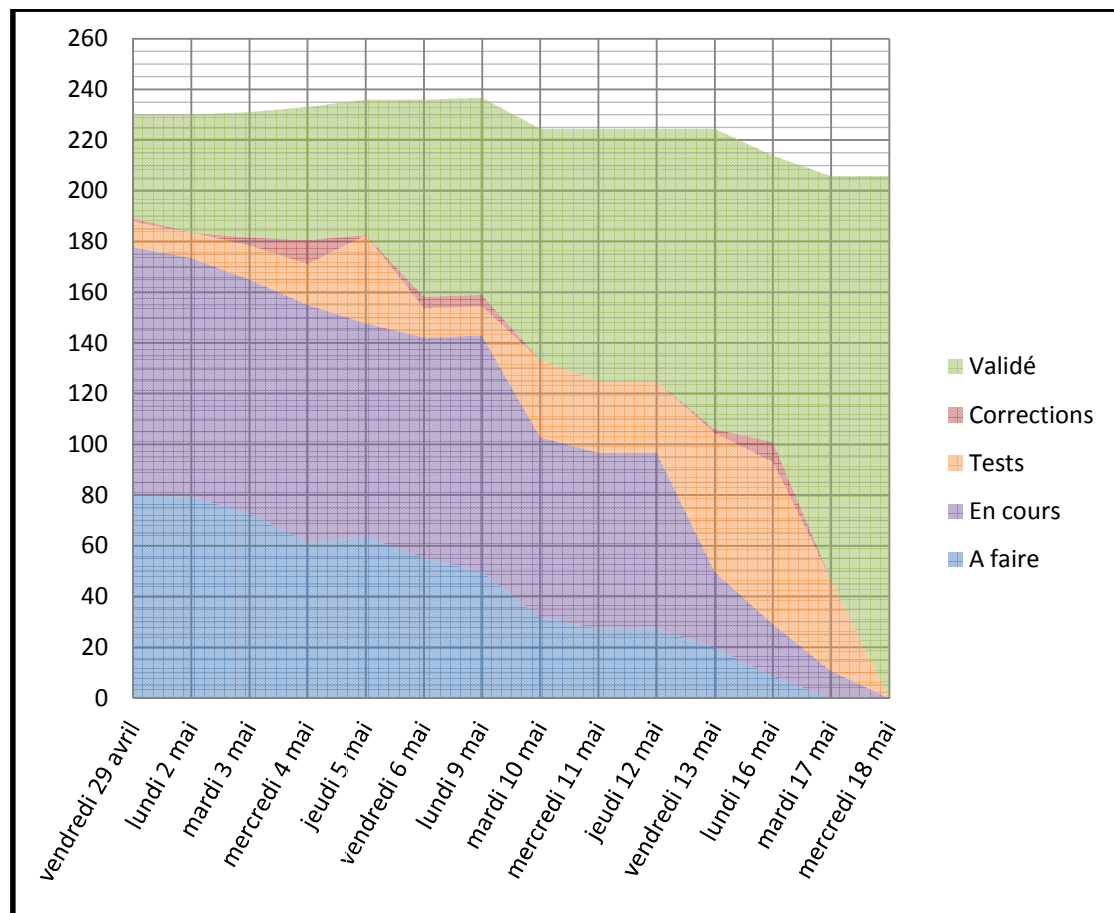


Figure 9: Graphic about where are all tasks

As I explained before, we have spread in a five columns blackboard all the tasks. On the first one, in **blue**, we have all the works that aren't started. On the next one on **purple** we have all the works that are on developing time. Once the developing time is finished they pass to be tested on **orange** color. When the quality team has finish of testing them, they will go to corrections column (on **red**) or to the last one (on green) that is the validated column.

The version started on the 13th April but the graphic starts on the 29th. This is because on the first two weeks there isn't so much valid information to read on that graphic; almost all the works will be on the first or second column.

As we can see on the testing part, the time when they have more work is the last days. For example the 16th May there were around 60 developed days. We know that this isn't very efficient but there aren't any reasonable methods to fix this. At first I thought that maybe was interesting to divide each big task in two or three individual smaller works but there are some of them that it's impossible. Also to estimate how to divide and the number of days for each work required a lot of work before start developing.

The last thing that is going to be explained about the agile methodology used at Travelsoft is about the estimation part. This is one of the most important parts of the version because this one isn't well done the version couldn't be finished. So, as my boss Mathieu has explained me, to estimate each task, it's important to divide it as much as possible. Once the task is cut, it's time to add the number of days for each subtask, normally no one has less than 0.25 because only the building time, the merged time the small part of developing is more than that. To finish the estimation there is some time to the corrections depending the risk that has each task.

To know how many days have each team to develop there is a formula inside the company as we can see on the figure number 10 on the page 26 that I found very interesting. Before selecting the tasks to do, the time of each developer can work on the version is calculated. That time are the days that each person will go working will be multiplied by a coefficient. That coefficient depends on the person. For example, I started on the first version with 0.6 because I didn't know so much about the reservation module and I lost time to find where I needed to make the change to fix the task. On the 8.3 I work with a 0.8 coefficient and to finish these last versions I am being working with 0.9. Until now everything it's normal. But for example my boss Mathieu has normally a 0.5. It means that only the 50% of the time that he is at the

company is going to work on the reservation module. This is because always there is people who make a lot of questions like for example the support team. Also he helps me every time I need something (now much less than before) and of course al the meetings that he has to improve the application.

| Affectation | Coeff | Nb days dispo | Nb working days |
|-------------|-------|---------------|-----------------|
| TLP | 0,8 | 24 | 19,2 |
| MO | 0,9 | 22,5 | 20,25 |
| MB | 0,5 | 22,5 | 11,25 |
| | | 69 | 50,7 |

Figure 10: Table about the coefficients

I explained this because as we can see on the figure number 7 the reservation team had only 31 working days and we are three on the team and I wanted to explain why. Lan that is the third person of the team worked in other project and if we add the 0.5 of Mathieu and my 0.9 with any holiday this is the developing days that we get.

Tools

In this internship, I am having the opportunity to use many open source tools. Indeed, the policy of Travelsoft is to use Open Source software [3] that is as good as the other ones.

All the code lines are done in Eclipse Environment [4]. This tool is really useful because a part of being Open Source, it has a lot of plug-ins that can make the programming faster and easier as for example check style, subversion, ant and more.

To manage all the different versions, changes, uploads Travelsoft decided to use Subversion [5]. Subversion is a free/open source version control system. It manages files and directories, and the changes made to them, over time. This allows you to recover older versions of your data or examine the history of how your data changed.

In Travelsoft, compilation of projects is carried out using the tool Maven [6] that can compile an application, pass the unit tests, generate many project reports and build deliverables (jar, war, ear, etc...) the times that we want. This tool operates from an XML description file. This descriptor defines the structure of the project links with other projects. Apart of that, Maven can use different plug-ins in its descriptor to execute. With this tool, we can have actualized our entire project each time that someone uploads a modification on the code.

Also there is another tool called Bamboo [7] that is used to manage all the continuous integration. This tool allows the automatic projects construction. When a developer sends the work that he has done to the remote working directory (SVN Repository), Bamboo will detect the source and automatically will build the project deliverable. If any problem or error is detected during the assembly, an email is sent to the developer who has committed the source. This tool is really interesting because it avoids big problems that normally can be when big projects are built.

When I arrived to Travelsoft Company we used only Bamboo but now we are using Jenkins [8] too. I don't know which one is better but for me Jenkins can be configured more easily. It has the possibility of duplicating the configurations that are done before and there the only thing that we have to change is the projects that we want to build after it and the name of the project, the other configurations will be the same.

Tools for managing tasks and anomalies are used also in Travelsoft. In these tools we can include Jira [9]. This one is currently internal tool that saved all

the different task of the live of any project with a lot of information like the status, importance, description, who is going to implement, etc... Jira is used extensively by the quality team who created a record when a behavior abnormal is detected. This tool is very useful in the method of development used by Travelsoft because they can be traced back problems fairly quickly. Moreover, since this tool uses Maven, reports are generated for each modification of the project and we can continue all the duration of the project during all the builds. Apart of doing the build when someone commits some code it is possible to scheduled the builds.

Those tasks are also added one by one on the table that is on the wall. There are different columns to post all the tasks (one post-it for each task) like: waiting, ongoing, testing, correction and validate. Even there are some graphics for each part of the project where you can find if the work that you are doing is on time, delayed or faster than predicted.

Formation

Parts of the project

The entire Travelsoft project is defined in a lot of smaller projects and each one has a boss. Some of them are connected among themselves. So because of that it is very important the communication between all the different team of the company.

Now I am going to resume one by one all the work that the different teams do at the company.

All of them are very important but one of the most important blocks is the **Import** bloc. This bloc is connected to the data base and at the same time to all the TOs. Time to time it ask to the different TOs information about the price, availability, information to add on the product information, etc... That information is contrasted with the information that is on the data base and if the information is not actualized, is overwritten.

To manage all the configurations, travels, users and a lot of more things, there is a project called **Back Office**. With this back office all the TD can configure everything for them. They can create a new travel packages, change the availabilities and the information of a package that after through the Import bloc will be actualized in our data base, so it is connected to the data base. Apart of that they can confirm or cancel some travels that they were already sold. To do all the cancelation and the confirmations they are not straight connected to the TOs, but they are connected to another project of Travelsoft. That is called Connectique and is in charge of connecting different internal projects with the TOs. In resume if we have to explain in

a unique line the Back Office we would say that is the interface that all the TD use to configure their information.

If we want to sell any project we need a **Catalogue** that will be the interface where all the clients will use to buy the travels. Is connected to the data base, where it takes all the information to fill in all the web pages. Also is the door to the project where I am working, the reservation.

On the **Reservation** project, as its name says, it is done the entire reservation of each travel package. It is one of the most important parts of all the sales process because if this part does not work, not the TOs neither the TD would earn money. One side is connected to the data base where for each reservation a new dossier is done and saved. Also it is connected to some extern systems like payment or insurance companies to get the price or confirm the price of each one. The last link is the link that connects the reservation part with the Connectique (that now will be explained) to ask for information to finish the booking correctly.

The last part is the part that we have told before twice and it is the **Connectique** project. This project is in charge of encapsulating all the information that comes from the Back Office or from the resa to send correctly to the TOs. Each TO have different method to connect to Travelsoft and with this project the TOs are divided in different groups depending the way that they work. With this part is easier to connect a new TO and to manage all of them. It is similar to the import project but with the difference that in this block all the transactions are done in real time, when the user is doing the reservation. On the contrary, on the import project, the transactions are done in a scheduled time table.

Reservation

This is the part where I am going to work for all my internship. Before I have made a resume about how the resa works and now I will explain how it works internally.

For each reservation we have 4 steps differentiated: rate details, inscription, payment and confirmation. All of them are important for the reservation project and now I am going to explain them one by one.

- Rate details: this is the first step. If we want to start all the reservation from here, we need an object called InformationData that after will be explained, with information about the travel. With that information and before opening the rate details page, a search request is done to the TOs to confirm if the information that comes in that object is available and if it is, a page with a lot of travel options will be opened. The content of the page will be different depending the travel because some of them comes with transport and other ones not and the same for the services that the travel has. Once the page is loaded with the information that we have received from the search, we can choose different thing about the travel, like, pension, category, transport, insurances, mandatory options, etc... of course if the travel allows us.
- Inscription: when the client is satisfied with the options and he have clicked on the next step a new request is done to confirm that all the information that the client wants is available and like this to do not have any problem after. If the services that the client wants are available a new page will open to fill in all the information about the travelers.

- Payment: when the client has filled in the information a new page is opened with all the different payment methods. The thing that the client has to do now is to choose a payment method and continue the payment with the correct values of the payment if necessary.
- Confirmation: this is the last step and of course the most important, because if we have any problem with the payment we can have big problems. Once the client has chosen the payment method a booking request is sent to the TOs to confirm that all the information about the travel that the client wants is available. Apart of that another request is sent to an external payment method if it's necessary to confirm the payment. When everything is OK, the book is done and we change information in our data base. To save in our data base, we change the availability of our travel package and we save a new dossier with the information about the travel.

Continuing with the resa I have to explain that not only all the resa are the three requests about I have spoken before. There is another important element which is the static content. In this content we can have different files like the Velocity Template (VM) files, java script files, the css, properties, etc... The advantage that we have with this content is that for each TD we can take as much files as we want from the generic specific content from our project depending requirements. With this project we could change easily the different language or titles (changing only a properties file), the colors, images, sizes, etc... (Changing the css), the position where each bloc of the page is going to be inserted (changing some VM files), and more.

On the static content there are different folders. In one side we have the folders for the clients that starts with the string "cs" or the internal folders that starts with "ts". On the second side we can check if the files are going to be managed with jBoss ("local") or with apache ("web"). To finish the naming there is a sandwiched "v"

which tells that the content is saved on subversion. Resuming, if we combine the different folders we will have 4, two for our clients and two for Travelsoft. On the web folders we are going to save files like some css, images, PDF files, java script files, etc... and on the local folders we are going to save xml files, VM files, properties files, etc...

It means that the internal project has a lot of different files that after can be changed or configured easily for each TD. This is done because there are a lot of TDs and everybody knows that each person or client always wants different things that another one wants.

For example, the first page of the reservation (rate details) can change for each TD depending the requirements of each company. Also if the same TD wants the same webpage in two different languages it will be necessary only to write three lines on the configuration file, and the page will be transformed to the English language too.

To better understand I am going to compare all the rate details page of Carrefour and Thalassa, which are among the most important TDs for Travelsoft.

I am going to start with the header; for Carrefour it's important to show a little resume of the travel and to Thalassa is not as important. If we want to configure this one for each TD, we have to add in the static content folder the VM file that configures the structure of the page and add or delete the blocs that we want. If the static folder is empty the web page for the client will fill in with the values by default that we have in our content static generic. On the figure number 11 of that we have on the page 34 we can see the header for Thalassa and on the figure number 12 on the page 34 is the header for Carrefour. As we can check on the figures the difference between them is really big.



Figure 11: Rate details header for Thalassa

| 1. RÉSERVATION | 2. COORDONNÉES | 3. PAIEMENT | 4. CONFIRMATION |
|--|----------------|-------------|-----------------|
| Récapitulatif de votre commande | | | |
|  | | | |
| Réservation Ville de départ : Paris Date de départ : jeu. 16 juin 2011 Date de retour : jeu. 30 juin 2011 Durée : 15 jour(s) / 14 nuit(s) Destination : INDE OPIL Inde Carrefour 15 jours / 12 nuits Voyageur(s) : 2 adulte(s) | | | |

Figure 12: Rate details header for Carrefour

For the bloc of lodging, transport and taxes we don't have as much differences as on the header bloc. On the figure number 13 on the page 34 you will find the first important bloc on the rate details page for Thalassa and on the figure number 14 on the page 35 you will find the rate details first bloc for Carrefour and compare by yourself.

| Détail et options de votre devis | |
|---|--|
| Forfait principal et taxes | 1 200,00 € |
| Hébergement | 1 000,00 € |
| Hôtel Hvar*** 1 x Chambre double (2 adultes) masquer Inclus | |
| Prestations | Pension Demi-pension : Inclus |
| Transport | |
| Aller Paris, sam. 28 mai » Split, sam. 28 mai | |
|  →  | Paris sam. 28 mai → Split sam. 28 mai vol spécial |
| Retour Split, sam. 4 juin » Paris, sam. 4 juin | |
|  ←  | Split sam. 4 juin → Paris sam. 4 juin vol spécial |
| Taxes | 200,00 € |

Figure 13: First important bloc of rate details page for Thalassa

Veuillez confirmer ou modifier votre choix

| | |
|--|---|
| Forfait principal | 3 340,00 € |
| Hébergement | 2 868,00 € |
| OPN Inde Carrefour 15 jours / 12 nuits Période : du 17 juin 2011 au 30 juin 2011 | |
| <input checked="" type="radio"/> 1 x Chambre double (2 adultes) masquer | Disponibles Inklus |
| Chambre 1 | |
| <input checked="" type="checkbox"/> OPTION CARTE FIDELITE * | C.FIDELITE/BALADE RICKSHAW : Inclus |
| <input checked="" type="checkbox"/> OPTION OFFRE RESERVEZ-TOT * | ECHARPE EN PASHMINA OFFERTE : Inclus |
| <input type="radio"/> 2 x Chambre simple (1 adulte) afficher | + 918,00 € |
| Transport | |
| Aller Paris - (Aéroport Ch. De Gaulle), jeu. 16 juin >> New Delhi, ven. 17 juin | |
|  Paris - (Aéroport Ch. De Gaulle)T-1 jeu. 16 juin | New DelhiDEL ven. 17 juin |
| Détails | |
| Retour New Delhi, jeu. 30 juin >> Paris - (Aéroport Ch. De Gaulle), jeu. 30 juin | |
|  New DelhiDEL jeu. 30 juin à 2h25 | Paris - (Aéroport Ch. De Gaulle)T-1 jeu. 30 juin à 9h55 |
| Détails | |
| Taxes | 472,00 € |

Figure 14: First important bloc of rate details page for Carrefour voyages

As we can see on the two images before the style between themes is different. Also we can find a different title for the first bloc; on the first figure is written “Forfait principal” and on the second “Forfait principal et taxes”. That is changed on the properties files.

For the bloc of services there are a lot of differences but these are because the travel package is very different. As you can see on the figure number 15 on the page 36, Thalassa has a lot of services on the travel package. On the figure number 16 illustrated on the page 36 too, you can find the services that Carrefour offers for his travel. As you can see the services that each one offers aren’t the same so the blocks added will not be the same too. The only thing that changes between them is the blocks that are going to be printed.

| | |
|--|--|
| Prestations | 84,00 € |
| Autres options | |
| Afin d'ajouter une option cliquez ci-dessous | |
| <input type="checkbox"/> PENSION COMPLETE/ FULL BOARD | 85,14 € |
| Assurances | 54,00 € |
| <input type="radio"/> Aucune assurance | 0,00 € |
| <input type="radio"/> Formule Annulation - Europe et Moyen Courrier masquer Conditions générales | 42,00 € |
| <input type="radio"/> Formule Assistance- Europe et Moyen Courrier masquer Conditions générales | 42,00 € |
| <input checked="" type="radio"/> Formule Multirisques - Europe et Moyen Courrier masquer Conditions générales | 54,00 € |
| Réduction | |
| Appliquer une réduction avec un code <input type="text"/> | <input type="button" value="Appliquer"/> |
| Frais de dossier | 10,00 € |

Figure 15: Service block on rate details page for Thalassa

| | |
|--|---------------|
| Prestations | 0,00 € |
| Prestations incluses | 0,00 € |
| Carnet de voyage | Inclus |
| Selon les contraintes du Tour Opérateur | |
| Assurances | 0,00 € |
| <input checked="" type="radio"/> Je ne souhaite pas assurer mon voyage <small>(Nous vous invitons à vérifier auprès de votre assurance personnelle la couverture d'une annulation éventuelle de votre séjour, les montants garantis, les franchises et exclusions appliquées ainsi que les conditions d'assistance et de rapatriement.)</small> | 0,00 € |
| <input type="radio"/> Assurance multirisque voyages étranger MONDIAL ASSISTANCE masquer <small>Formule multirisque : Annulation optimum - Dommages aux bagages - Assistance aux personnes - Interruption de séjour - Interruption activité de sport ou de loisir - Responsabilité civile vie privée (hors France) - Capital accident - Retard avion ou train - Assistance complémentaire aux personnes</small> <small>Conditions générales de vente pour Assurance multirisque voyages étranger MONDIAL ASSISTANCE</small> | 141,95 € |

Figure 16: Service block on rate details page for Carrefour

The total block is different too. Thalassa doesn't want to show as much information as Carrefour that shows a button to go to the step before and the availability of the travel. On the figure 17 on the page 37 you will find the total block for Thalassa and on the figure 18 on the page 37 you will find the total block for Carrefour.

| | |
|--------------------------------|-------------------|
| Total | 1 284,00 € |
| Toutes taxes incluses | |
| Étape suivante | |

Figure 17: Total block on rate details page for Thalassa

| | |
|--|--------------------------------|
| Montant total du voyage | 3 340,00 € |
| Toutes taxes incluses | |
| Le statut de votre commande est : Disponible | |
| Étape Précédente | Étape suivante |

Figure 18: Total block on rate details page for Carrefour

To finish with the rate details bloc, Carrefour wants to show a footer part on the page that is illustrated on the figure 19 on the page 37 that Thalassa doesn't wanted.

| |
|--|
| Pour toutes demandes particulières, selon disponibilité (ex : lit bébé, chambres communicantes, hébergement multiple, ...), merci de nous contacter au  0826 822 822**  |
|--|

Figure 19: Footer on rate details page for Carrefour

Data structure

For all the reservation part there is done a global data structure. It is done with XML Schema and after is parsed to a java classes (java binding) with the JAXB [10] project.

What is Java Binding?

I am going to explain this part because I think that is important to know how the base of the reservation project. JAXB is the acronym of Java architecture for XML

Binding. An XML schema is really easy to understand also for a person who doesn't know a lot about computer engineering and in this is based JAXB.

Finally the goal of the JAXB project is to develop and evolve the code base for Java Architecture. Resuming it makes use of a given schema of an XML document and automatically generates the required Java classes, corresponding to that schema. With this it will be easy to pass from a class to an xml file or vice versa. Also it important to tell that there is a small guide to declare the attributes or elements on the XML element but it's really simple, that it makes more interesting a JAXB structure.

On the figure 20 that we can see below it's explained with a graphic how works JAXB.

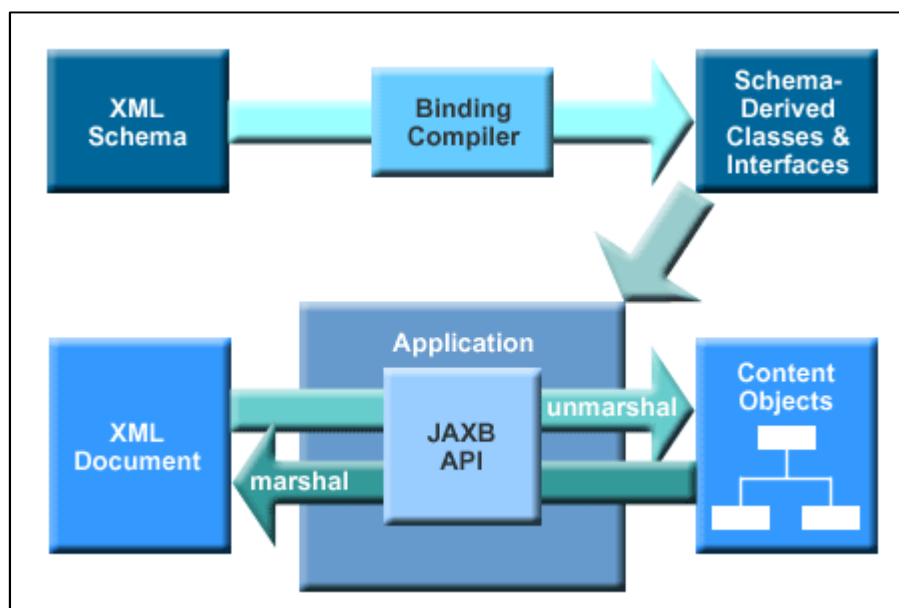


Figure 20 : JAXB schema [1]

So once we have all the classes created by the binding compiler there will be ready to be used on our application. After this we will need to marshal or unmarshal the objects depending what we want to do. The action of transforming the xml document in a java object is called unmarshal and the contrary action marshal.

For example if we want to create an object with from an xml we will have to declare a JAXBContext objet first with the object that we want to unmarshal as is figured on the figure number 21 on the page 39. After that we will create an umarshaller from the context to control the process of marshalling. To finish the unmarshall we have to call the marshal function with the data of the xml file like for example in an Input Stream and we will have a JAXBElement that we could be cast in our element without any problem.

```
public <T> T unmarshal( Class<T> docClass, InputStream inputStream )  
    throws JAXBException {  
    String packageName = docClass.getPackage().getName();  
    JAXBContext jc = JAXBContext.newInstance( packageName );  
    Unmarshaller u = jc.createUnmarshaller();  
    JAXBElement<T> doc = (JAXBElement<T>)u.unmarshal( inputStream );  
    return doc.getValue();  
}
```

Figure 21 : Unmarshal function

On the other side we can do a marshall defining the JAXBContext and the Marshaller. After them we have the possibility of declaring come properties like for example if we want to save formatted the data on the file. To finish we have to tell to the context what and where we want to save.

The other transformation is almost the same of this one but choosing the xml like an input and the java object like an output.

All the data is stored in an element called InformationData that has 4 elements:

- ReservationInfo: all the data that we need about the voyage. From the type of voyage to the number of person and their names.

- GlobalConfig: some global configuration information, as the contact emails, where is the reservation context, the channel that we are using to do the reservation, etc.
- SessionConfig: information about the configuration that we are using in the moment of the reservation. For example we will save information about the vendor, different payment methods, etc.
- SystemInfo: it will be interesting to save the status of the processes, the step where we are in each moment, codes about the channel and about the organization that is selling the package, etc.

Between them I am going to work a lot with systemInfo. There is all the information about the hotels, transports, services, etc... there and because I think that is one of the most important element, on the figure number 22 on the page 40 there is the code of the element, which will be useful to understand the data structure:

```

<xs:complexType name="ReservationInfo">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="status" type="Status" minOccurs="0" maxOccurs="1"/></xs:element>
    <xs:element name="vendor" type="Vendor" minOccurs="0" maxOccurs="1">
      <xs:annotation>
        <xs:documentation>The traveller who has placed this order.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="booker" type="Booker" minOccurs="0" maxOccurs="1">
      <xs:annotation>
        <xs:documentation>The traveller who has placed this order.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="traveller" type="Traveller" minOccurs="0" maxOccurs="unbounded"/></xs:element>
    <xs:element name="vacationPackage" type="VacationPackage" minOccurs="1" maxOccurs="1"/></xs:element>
    <xs:element name="costPrices" type="CostPrices" minOccurs="0" maxOccurs="1"/></xs:element>
    <xs:element name="payment" type="Payment" minOccurs="0" maxOccurs="1"/></xs:element>
    <xs:element name="reservationProperty" type="ReservationProperty" minOccurs="0" maxOccurs="1"/></xs:element>
    <xs:element name="productInfo" type="ProductInfo" minOccurs="0" maxOccurs="1"/></xs:element>
    <xs:element name="criteriaInput" type="CriteriaInput" minOccurs="0" maxOccurs="1"/></xs:element>
    <xs:element name="contextInfo" type="ContextInfo"/></xs:element>
    <xs:element name="bookingProcessReport" type="BookingProcessReport" minOccurs="0" maxOccurs="1"/></xs:element>
    <xs:element name="history" type="History" minOccurs="0" maxOccurs="1"/></xs:element>
    <xs:element name="searchRetrialData" type="SearchRetrialData" minOccurs="0" />
    <xs:element name="bookingRefs" type="BookingRefs" minOccurs="0"/></xs:element>
    <xs:element name="changeManagemet" type="ChangeManagemet" minOccurs="0" maxOccurs="1"/></xs:element>
  </xs:sequence>
  <xs:attribute name="sessionId" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="conversationId" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="changeStatus" type="ChangeStatusDef" use="optional"/></xs:attribute>
</xs:complexType>

```

Figure 22 : ReservationInfo element example

First days

As I explain before, the first part of the internship was the most boring part but at the same time the most important too. If I had not learnt the first two weeks all the material that Mathieu Bonin gave me, I would have taken much longer.

Once I learnt more or less how works the reservation part on the company they gave me a small “mini-projects” to do. With these small projects I learnt more because I was fighting with the java code, so was good to start little by little.

For do the different reservation tests, exists a web page where you can choose the different values of the travel and simulate the booking. The problem with this application is that for a lot of travel packages the software has to call to the different TO to ask the different data as for example the dates cost of the travel if we change something, etc...

Each time that we send a question for each TO we have to wait some seconds and that can be boring if you are testing. So the first part consisted in reading all the data from a XML file and open one of the 4 steps of the reservation really faster.

After that they told me that also will be interesting to add a function for each step, to save all the information that we have there in a XML file. After we could use the file that we have saved with the functionality that I have explained before to open all the steps of the reservation.

Apart of that we add some more functionalities as a function to display all the data that we have on the session about the reservation, a new attribute on the global configuration, a login to activate this attribute, etc...

Developing work

On the resa, it is very important to have any test environment; it means that to test every change on the project an entry point is needed. For that exists a page called testResa that gives to the reservation project the same information about the product to reserve that any TD has on their own page, so resuming we can say that the test resa page simulated the catalog.

The problem of doing all the tests is that you must call the TOs. Every time that we do a test with a product, we have to call the TOs asking for availability, price, etc... and that is not very efficient.

My first work at Travelsoft is closed related to this template. If I had to resume in one sentence I would say that my work here is to improve the way of doing tests. For that I have not only done a testResa without calling the TOs but also I improved a new bloc on every reservation pages to know the content of the element informationData and to save this content in a xml file.

Explaining all together is complicated, so I thought it appropriated to break down the explanation in a few main sections that will be explained below. The first work that I did was to add a new bloc on each step of the reservation. To add that bloc I created a new JSP page where I printed some information.

Version 8.1 and 8.2

When I arrived to the company as I have explained before, there was another methodology than the one that we have now. All the versions were longer and weren't as controlled has the new ones. The first version where I worked was the 8.1

and as on the next version I worked too on the same subject I am going to explain them on the same bloc. This was from my arriving day until July 12th.

Debug bloc

As I have explained before, I created a new bloc for all the steps. The bloc is the same that we can see on the figure 23 on the page 43. That figure has three different columns with different functionalities.

The screenshot shows a web interface titled "DEBUG TOOLS". It contains four main sections:

- Profil data information:** Contains three links: [Information data](#), [Configuration](#), and [Clean cache](#).
- Received profile:** Contains a list of profile data:
 - defaultResConfig:
 - channel: RESFORM
 - organization: org1
 - user: user1
- Canonical profile:** Contains a list of profile data:
 - defaultResConfig:
 - channel: RESFORM
- Generate XML page:** Contains a text input field for "File prefix:" and a "Save" button.

At the bottom of the interface, there is a status bar that reads: "bookingprocess : 'prerecette-front' instance on: 'baltimore' - 8.2-TEST - Built on 20110324-1143 by team."

Figure 23 : Debug block

On the first column we can check the value that has in every moment the informationData session variable or if instead I the data that interested you its only the configuration, you have another link to show. For both there is another JSP file with the value of a container that shows our request as you can see on the following figure.

On the middle of the block, we have two columns with the information about the profile. On the first one there is the profile information which we have asked the reservation and on the second one we have the profile with really we are doing the reservation. The profile information is information that tells us the channel and the configuration that we want to use during all the reservation.

To finish with the bloc, there is implemented very useful action. If you click on the Save button, a new xml file will be created with the values that are on the InformationData object in that moment, it means that if you click on the button while you are on a rate details page, you will save only that information and not the information about the travelers or payment because in that moment won't be available. There is the opportunity to add a prefix for all the files that we are going to save because if we don't do this and if we save a lot of files, it will be very difficult to find the file after.

The files that are saved are very important to simulate after all the reservation without calling the TOs with the other application that I did and below I am going to explain. With the combination of the two functionalities we will be able to do all the tests faster than without them.

To be more efficient, we have decided to configure a constant with the directory where all the files are going to be saved. The repertory will be on the static content in a *saved* called folder. As the name of our files it's a little bit illegible because we add the timestamp, the user has the opportunity to add a prefix to the name of the file. With that prefix, if we have a lot of files in our folder it will be easier to find them after.

As you can see, this application is only interesting to our interests and maybe to the interests of some TDs but not for all the clients that are going to buy any travel. A result of that is the configuration attribute that is created on the configuration file. There is an attribute called enableDebugMode that has a Boolean value and if this attribute is not enabled the debug bloc won't be added.

There is another method of activating this attribute and is with a function called from struts configuration. It is possible to call the function from your browser to activate the debug mode. This action that I have explained just before, it is a little bit dangerous because if any one called by mistake or because they want to do bad

things, they can activate the bloc calling the action. For correct that I have added a login where one user and one password will be asked. These passwords are configured in an internal file.

Reservation step from an xml file

Continuing with the xml files and with the last part that I have explained before, I made another function that permits to open any step of the resa without calling the TOs. For this application you can use any xml file with an InformationData element saved.

To read all xml files I used the JAXB element class. The goal of JAXB is to develop and evolve the code base for the reference of JAXB, the Java Architecture for XML binding. So with this class if the XML file is correctly built comparing with the elements that we have in our intern project, the element will be built without any problem. It is important to know that this JAXB project it is really sensitive and if the element it isn't exactly as the element that we have in our project, we will have an error.

The step that we are going to open with this file is not the only thing that we can choose. Apart of that we can select the configuration that we want to use while we are building the corresponding webpage. This is made because the versions go on and maybe the configuration can changes, so if you have in your machine the ongoing version of the resa, on the xml file you have an old one and if you try to build the page with the configuration that the file has, you can have undesirable results. On the figure number 24 on the page 46 we have the interface where you will be able to simulate the reservation step from a file.

Figure 24 : Reservation step from a file

Email sender

Another interesting tool is the possibility of sending all the different emails without doing the action that sends these emails. There are 19 subjects with 4 types of different clients. If we want to test all of them after doing any change on the code could be really hard and expensive. With the send mails tool already it is possible simulate all the emails with their different data and configurations.

For example for the email of quotation, it is necessary a PDF file where is written everything related with the rate details page and the options that each one has chosen. On the figure number 25 on the page 46 we have the interface where you will can simulate all the possible email to send.

Figure 25: Email sender interface

All the emails are generated with VM templates. There is a general template that calls to other ones depending the type of email that is going to be sent. So thus there is possible to configure the email to the requirements of each TD. As for the different pages that the resa has, the messages are saved in different properties files. In this manner there is a chance of the multi langue, because only changing the string on the properties file you can have all the email or the page changed.

Rate details generator

The last work that I have done at Travelsoft is a rate details generator. The rate details page has different blocs with information about the reservation that the client wants to do. The entire page is divided in two big blocks. On the first one we have information about the lodging and the transport and on the second part everything about the services.

On the first side there are three sections. You can see the lodging with the category, pension and options if there are, also if the trip has transport there will be the different transport to choose and to finish there are different taxes for the travel that the client has chosen.

On the second side we can see the different services that a travel has. On the top of the services bloc there are the options, mandatory and optional. After that there is information about the renting, if there is a choice. Continuing with services, we can select the delivery type and the insurance type. To finish with the bloc there is an option to insert a reduction code and another one to the file fees that every TD can insert a quantity.

Although all the package travels have the lodging bloc, they don't have all other always. It is important to know that normally the most used blocks are the

lodging block, the transport to all the international travels and the mandatory options because there are travels that have options like any boat trip that are on the cruises.

The motive of the rate details generator template is to select the different options that the user wants and show the rate details page with those options without calling any TO. With this application we can show to all the clients how works our reservation first page and how their products will be sold, easily when they come to us asking if we can sell or not their products.

Now is possible to configure the reservation with all the different configurations that you want without the necessity of searching a trip with the characteristics that we want. The application reads from an xml files all the information. There is an xml file with the informationData element with the vacation package empty. Then depending the configuration that the user wants, different xml files will be read and cloned if the quantity of the element that the user wants required.

For this tool, different xml files are needed, as many as number of combinations for each functionality. We will save them in a folder on the server to have an easy access. For all this work there is another file that is called generatorConfiguration where all the configurations that are allowed for our template generator are defined. For example, if we want to look for any simple lodging with options and category we have find a line in our code like the line that is on the figure 26 on the page 48.

```
<functionality type="lodging" fileName="/var/data/tmp/files/Lodging/lodging_1.xml">
  <factors>
    <factor>withCategorySimple</factor>
    <factor>withOptionsSimple</factor>
  </factors>
</functionality>
```

Figure 26: Functionality in the config file for template generator

The configuration file for this tool has as much functionalities as files on our xml data base we have. As we can check on the example before, the functionality type is lodging; we also have the path of the file which has the two factors are defined.

So that everything is correctly read, we have added a new xsd file to our elements. There is a global element called infoContainer that has a sequence of functionalities. Each functionality has two attributes that are file name and functionality type; also there is another element (Factors) that has all the different options that has the file. The last element is the element Factor who has as many options as the file has. If there is any doubt, below, on the figure 27 on the page 49 is shown a figure with the element.

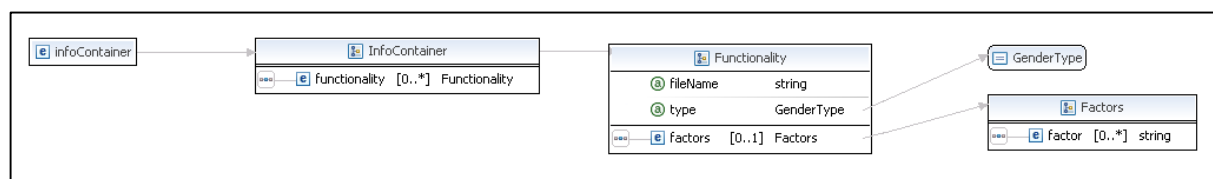


Figure 27: Info container element structure

The gender type that you can see on the type attribute is an enumerator that as a string value for each different type that we has on the rate details page. On the figure 28 on the page 49 is printed all the possibilities that there are.

```
<xs:simpleType name="GenderType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="lodging" />
    <xs:enumeration value="transport" />
    <xs:enumeration value="taxes" />
    <xs:enumeration value="optionalservice" />
    <xs:enumeration value="mandatoryservice" />
    <xs:enumeration value="filefees" />
    <xs:enumeration value="deliveries" />
    <xs:enumeration value="insurances" />
    <xs:enumeration value="reductions" />
    <xs:enumeration value="rental" />
  </xs:restriction>
</xs:simpleType>
```

Figure 28 : Gender type for the functionalities on the template generator

On the template generator template there two important files to read that are the information data template that we have explained before and the configuration file. As we are going to work with this files while the project is in production it is important to save them on the server. For that we have decided to save both with all the small files with different combinations of options for the rate details page on the static content. These files are for us so the folder where we are going to save them is `tsvlocal` and there we are going to create another one with debug name. With this configuration we could use the template generator in production without using any file that is physically in our computer.

As I have spoken a lot about the template generator but maybe you don't have the idea that how can be built the interface, I am going to simulate a template and on the figure 29 on the page 51 we can find the interface to start the simulation.

Package

☒ Lodging

☒ Simple ☐ Multi

☐ With Options
☒ With Pension
☒ With Category

☒ Transport

☒ Simple ☐ Combination

☐ Pre
☐ Post
☐ Intern
☒ Normal

☐ Taxes

☐ details

Services

☒ Mandatory

☒ Type A x module(s)
☐ Type B x module(s)
☐ Type C x module(s)
☐ Type D x module(s)
☐ Type E x module(s)

☒ Optional

☒ Type A x module(s)
☐ Type B x module(s)
☐ Type C x module(s)
☐ Type D x module(s)
☐ Type E x module(s)

☐ Rent a car

☐ Type A x module(s)

☒ Insurances

☐ Intern Insurances
☒ Europ Insurances
☐ TO Insurances

☐ Deliveries

☒ Reduction

☐ File Fees

InformationData template path
Generator configuration file path

or

Bookingprocess : 8.2-TEST built on 20110324-1143 by team.

Figure 29: Rate details template generator interface

With the configuration before we are going to build a rate details page with a lodging who has pension and category. After that, we want a normal transport, that means the going and come back from the origin to the destination. For the options we have choose two mandatory options from the first type that are fixed options (without the possibility to choose anything, not the quantity neither the date). We are going to add too an optional service to look the difference between the optional and mandatory services. To finish the rate details page we are going to incorporate an extern insurance bloc which is an insurance block that Europ assistance insurance gives us and a reduction block.

There is the possibility too to change the information data template selecting the path on the static content for the file and the same for the configuration file. It is important to know that the information data template is not a normal one and the configuration file it is difficult to change if you don't know so much about the subject. That is why is not recommended changing those files.

On the figure number 30 that is on the page 53, you will find all the result that we had when we did this configuration for the template generator interface.

Veuillez confirmer ou modifier votre choix

Forfait principal

HÉBERGEMENTS ET TRANSPORT

Hébergement
 Product name from testResa
 Période : du 10 juillet 2011 au 17 juillet 2011

☒ 1 x Chambre double (2 adultes) [masquer](#) Disponible Inklus

Prestations Pension DEMI-PENSION : Inclus

Chambre 1 Catégorie Chambre double BAIN, minimum : 1 adulte age maxi enfant=12 : Inclus

☐ 2 x Chambre simple (1 adulte) [afficher](#)

Transport

Aller PARIS ORLY OUEST, dim. 10 juil. >> Tenerife (Aéroport Reina Sofia), dim. 10 juil.

| | | | | |
|--|--|---|---|---|
| | PARIS ORLY OUEST dim. 10 juil. à 20h10 | → | Tenerife (Aéroport Reina Sofia) dim. 10 juil. à 23h30 | Disponible |
|--|--|---|---|---|

Retour Tenerife (Aéroport Reina Sofia), dim. 17 juil. >> PARIS ORLY OUEST, dim. 17 juil.

| | | | | |
|--|---|---|--|---|
| | Tenerife (Aéroport Reina Sofia) dim. 17 juil. à 14h20 | → | PARIS ORLY OUEST dim. 17 juil. à 19h15 | Disponible |
|--|---|---|--|---|

Prestations **-47,90 €**

PRESTATIONS OBLIGATOIRES **-108,00 €**

| | |
|-----------------------------------|----------------|
| Assistance Rapatriement | |
| Rapatriement Monde incluse | Adulte(s) 2 |
| Assistance Rapatriement 2 | |
| Rapatriement Monde incluse | Adulte(s) 2 |

OPTIONS

☐ Assistance Rapatriement

ASSURANCES **110,10 €**

| | | |
|---|------------------------------|----------|
| <input type="radio"/> Aucune assurance | | 0,00 € |
| <input type="radio"/> Assurance annulation 1 masquer | | 110,10 € |
| Annulation | 7 500 € Maximum par personne | |
| <input checked="" type="radio"/> Assurance annulation 2 masquer | | 110,10 € |
| Assistance avant le voyage | Classique | |
| Assistance et rapatriement | Classique | |
| Assistance après rapatriement | Classique | |
| Protection juridique | 3 000 € Maximum | |
| Annulation | 7 500 € Maximum par personne | |
| Interruption de séjour | 7 500 € Maximum | |
| Bagages et Effets Personnels | 1 500 € Maximum | |
| Accident de Voyage | 30 000 € Maximum | |
| Responsabilité civile | 4 000 000 € Maximum | |
| Ratage d'avion | Nouveau billet d'avion | |

Veuillez trouver les [conditions générales de vente](#)
 Vous êtes éligible si vous êtes résident en France

RÉDUCTION **-50,00 €**

Total **50,00 €**

[Étape Précédente](#) Étape suivante

Figure 30: Rate details result for the configuration of the figure 22

Diagrams of the 8.2 version

Unless the work that I did on the 8.2 version hasn't got too much to express on different diagrams I found interesting to do because is the part where I work a lot and the part that I more know. Also I think that a report won't be finished until some diagrams are done because is one of the most important part of the computer engineering.

I decided to show on this bloc of the report a use case diagram, a class diagram, an activity diagram and a deployment diagram. A sequence diagram will be show in other section because I found interesting the sequence diagram that we can have with a payment method.

Use case diagram

For these four blocks that I have developed, we can't have too much use case and I have decided to create two simples to understand everything easily. On the first one that is on the figure 31 on the page 55 I have added the global case. On the second use case that is the figure number 32 on the page 55 too all the functionalities are added and I don't do more because each one hasn't got more functionalities.

About the users, it's clear that we have a global user that can be any designer, developer, person from quality team or product team and because of that I decided to divide them. Normally each mini group of people will use different projects but all of them are accessible.

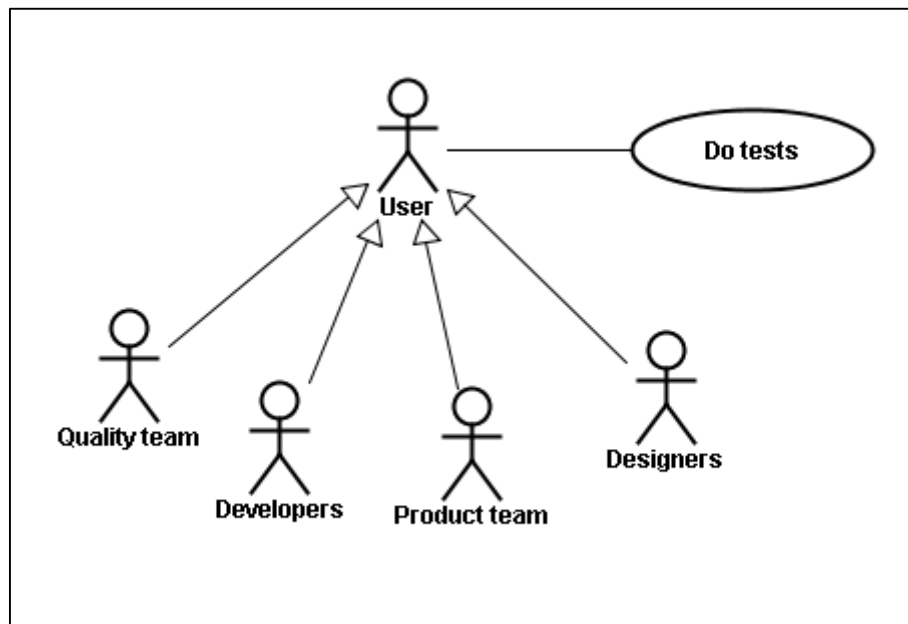


Figure 31 : Do tests use case (depth 0)

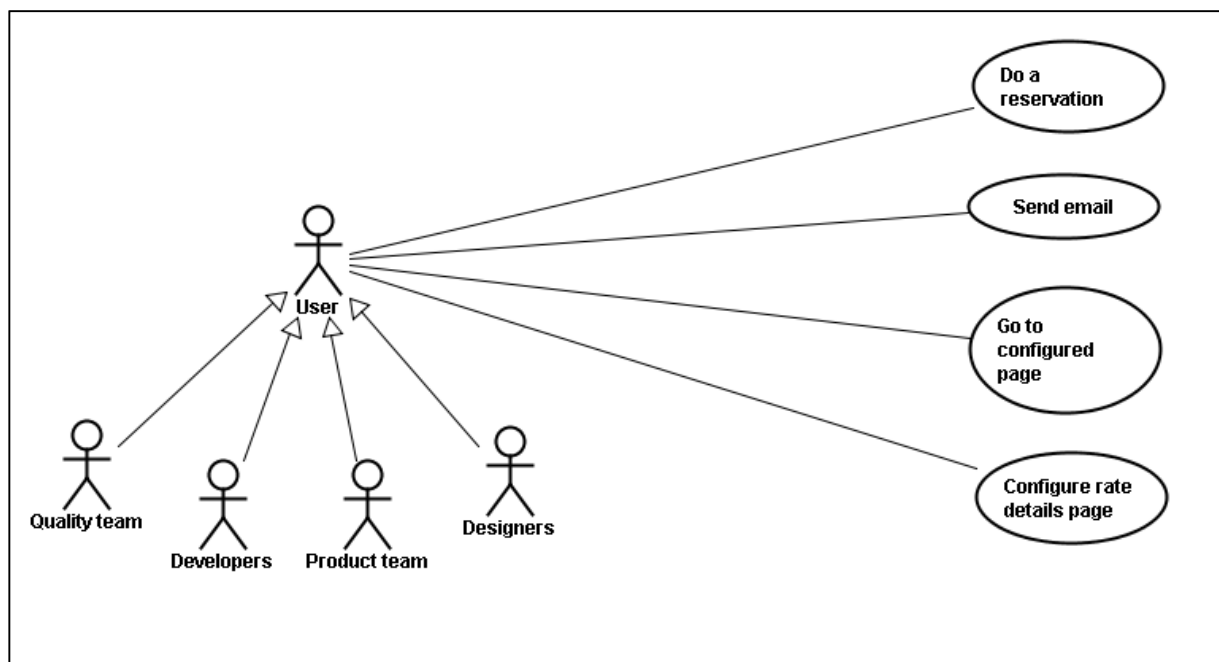


Figure 32 : Do tests use case (depth 1)

Class diagram

This is the most important diagram for me because it shows where is saved all the information during a reservation. As we can see on the figure number 33 that is on the page number 56 all the classes that I developed inherit from a global class called Abstract Session Action that at the same time inherits from Abstract Base Action.

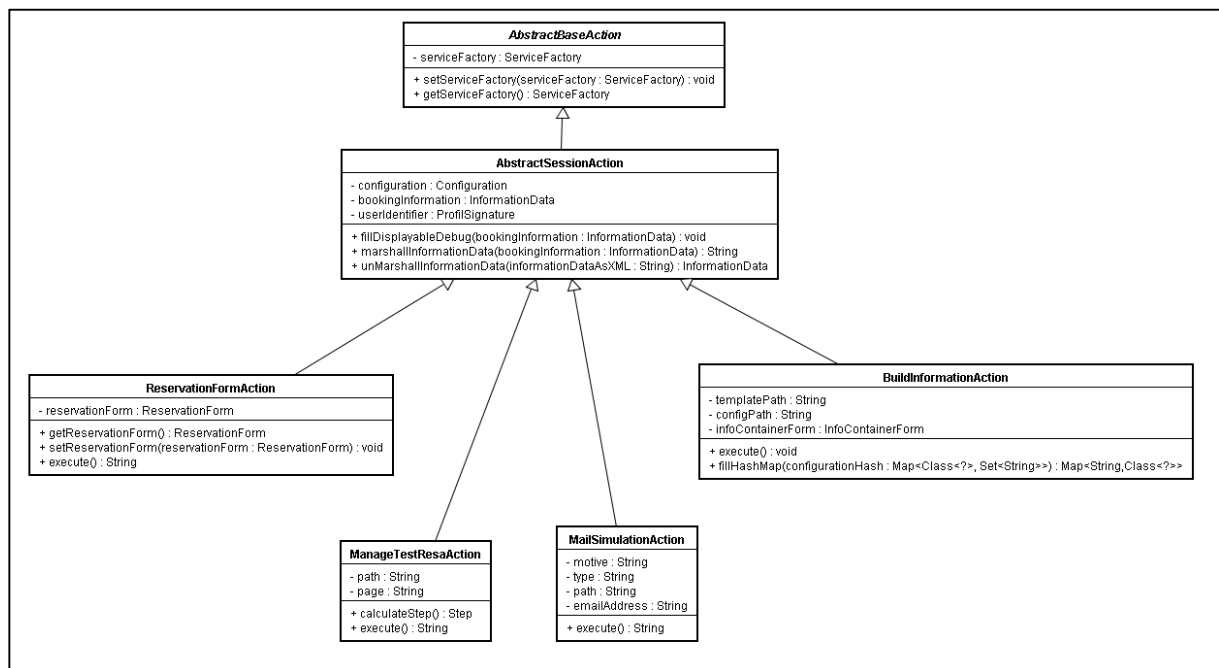


Figure 33 : Class diagram of my work in 8.2 version

The abstract base action we can call like the mother class. It's a class that saves on the service factory a lot of information. Service factory class is the class that manages all the different type of cache that is on the reservation method. It can save from a properties message to a path access to a velocity file. There are simple things that make really faster the reservation of a travel.

In other place we can find abstract session action and is the class that saves everything that has any relation with the session that you are managing. For example

all the information about the trip that you are going to do is on the variable information data. The configuration about the reservation is saved in other one and the most complex that is the user profile is in other one. For me the user profile is the most complex variable that we can find in this class because it's the variable that saves the encoded signature of every user and at the same time the session ID. With the session id we are capable of having two reservations from the same IP and this is really interesting.

Activity diagram

I try to resume as best I could the different states of a reservation method in the activity diagram that we can find on the figure 34 that is on the page 57.

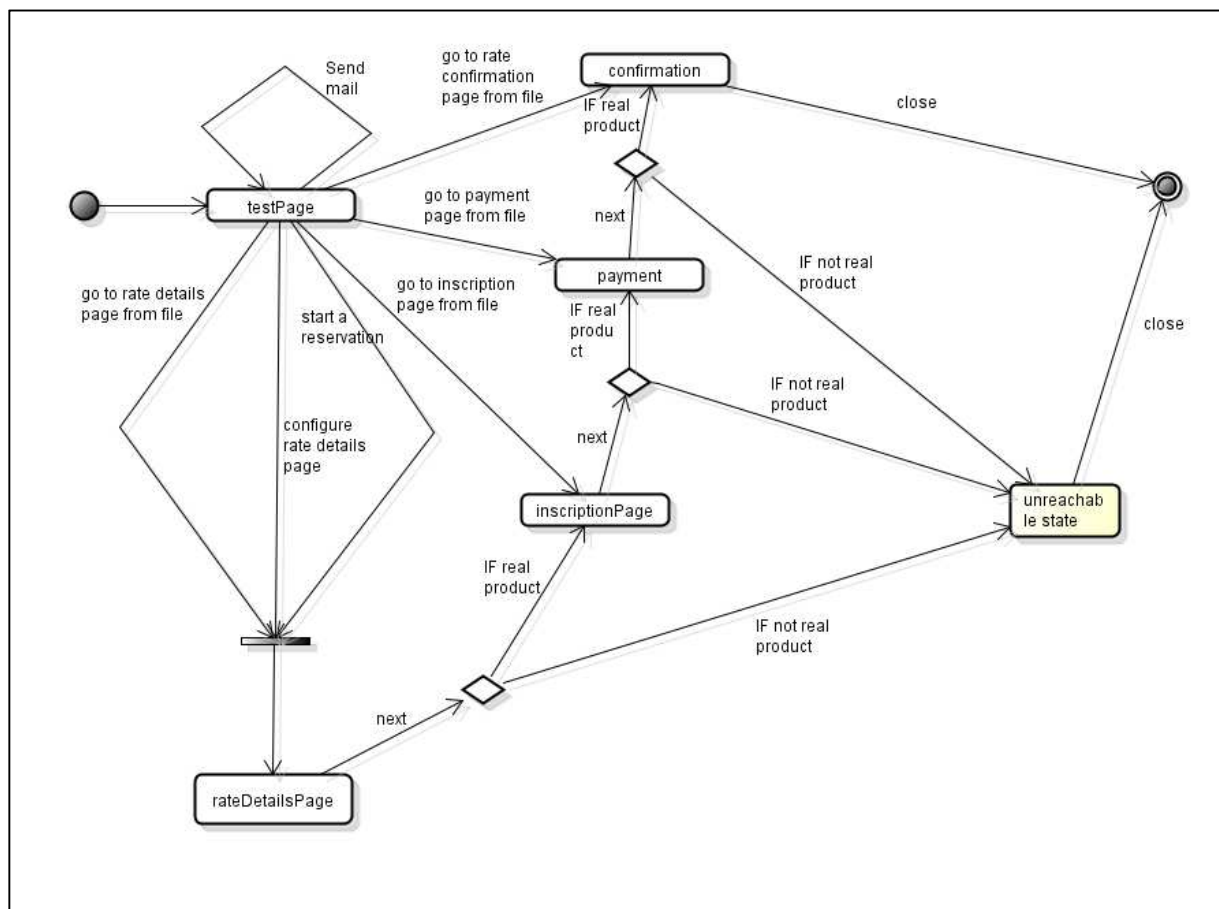


Figure 34: Activity diagram of my work in the 8.2 version

The start point is the test resa page. On that page we are capable of doing a normal reservation, sending an email, going to any page of the reservation reading the information from a file and also we can configure the rate details page as we want.

As we can see from the test page we can go to the rate details page from three different methods but later we couldn't be capable to continue the reservation depending from where we have start. For example if we start the reservation from the simulator that builds the rate details page from information that takes from simulated files, we won't be possible to verify with a quote the price and the availability. This is because between rate details page and inscription page a request is done to the tour operator to verify the prix and the availability that won't be possible to do simulating the data because each one will come from a different TO and the TO won't have the enough information to answer us. If this happens we will arrive to an unreachable state.

As we can see on the diagram there is the possibility of going until any page from a file but isn't configured to continue because when I did this wasn't the goal. I want to remark that when we click on the back button we have the same actions as with the next button but I didn't printed there because the diagram wasn't nice with so many lines and because of that I have decided to explain here.

Deployment diagram

Until now how a project was deployed to use it wasn't in my interest, from when I started my internship until now it started to draw much attention. As the entire diagram is too big I have decided to divide it in three different images, one for each server of a client. On the figure 35 that we can find on the page 59 we can see the deployment diagram of front office projects that are inside Travelsoft.

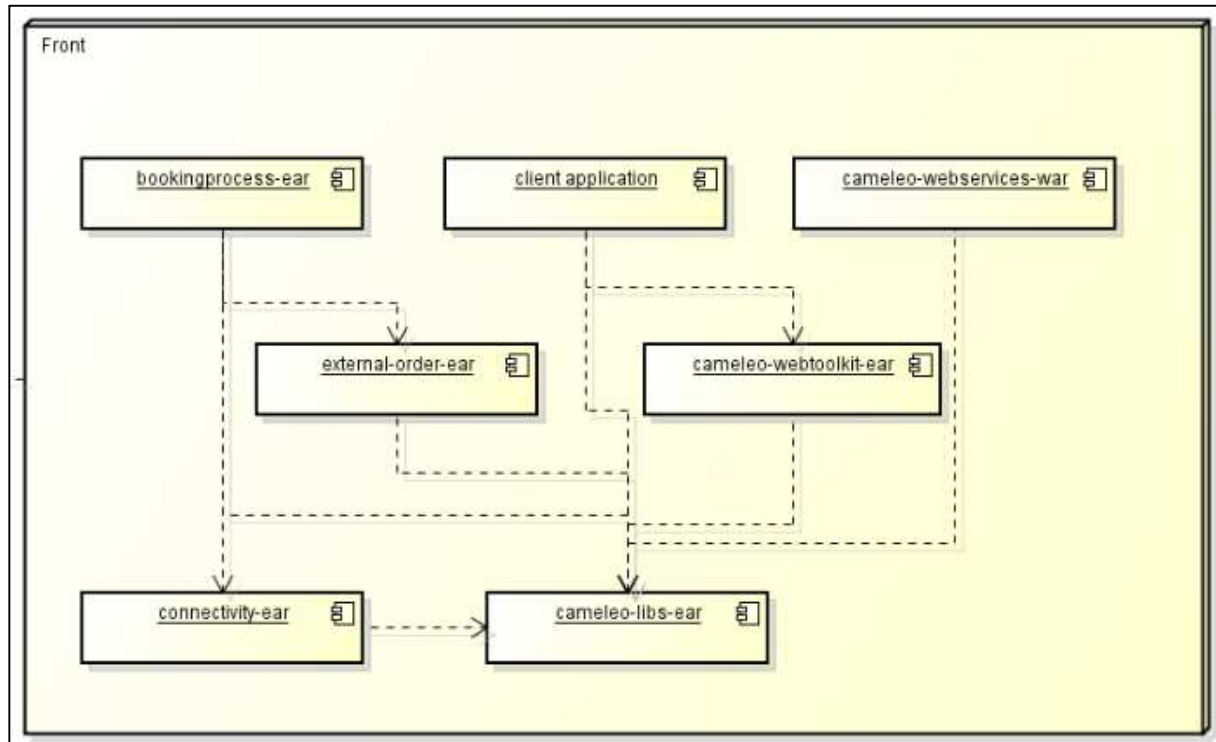


Figure 35 : Deployment diagram for the front office server

As we can see on the diagram to build the project that Travelsoft created for one client it's more complex than it appears at first. I am not going to explain a lot of this part because I only worked on the *bookingprocess* and *connectivity* projects and I don't know too much about the other. So because of that I am going to resume everything. We have a global project that everybody needs to be built that is *cameleo libs*. It has a lot of functions that a lot of projects have in common, we can see this like an API. Also there is a *connectivity* project that makes all the connectors between our software and the different tour operators.

On the middle part we can find two projects called *external-order* and *cameleo-webtoolkit*. The first one is used to save different book dossiers and to connect the *reservation* project with the back office; it means that once a reservation book is done, this project will do some treatments before sending the information to the back office. The second is used like an API for the *client-application*.

On the first part we have the projects “accessible” by clients. On the bookingprocess we can find a lot of things where I work, like for example the interfaces that we can see while we are doing a reservation and all the treatments that are done behind this to be possible to book a trip. On the other hand we can find *client-applications* and *cameleo-webservices* where we can find for example the catalog that the client can use to search different trips.

Once I have explained the part more accessible by the client I want to explain the back office that it's really important too. There we can find a lot of configurations and also configure different products among many other things. On the figure number 36 on the page 60 we can find the deployment diagram for the back office.

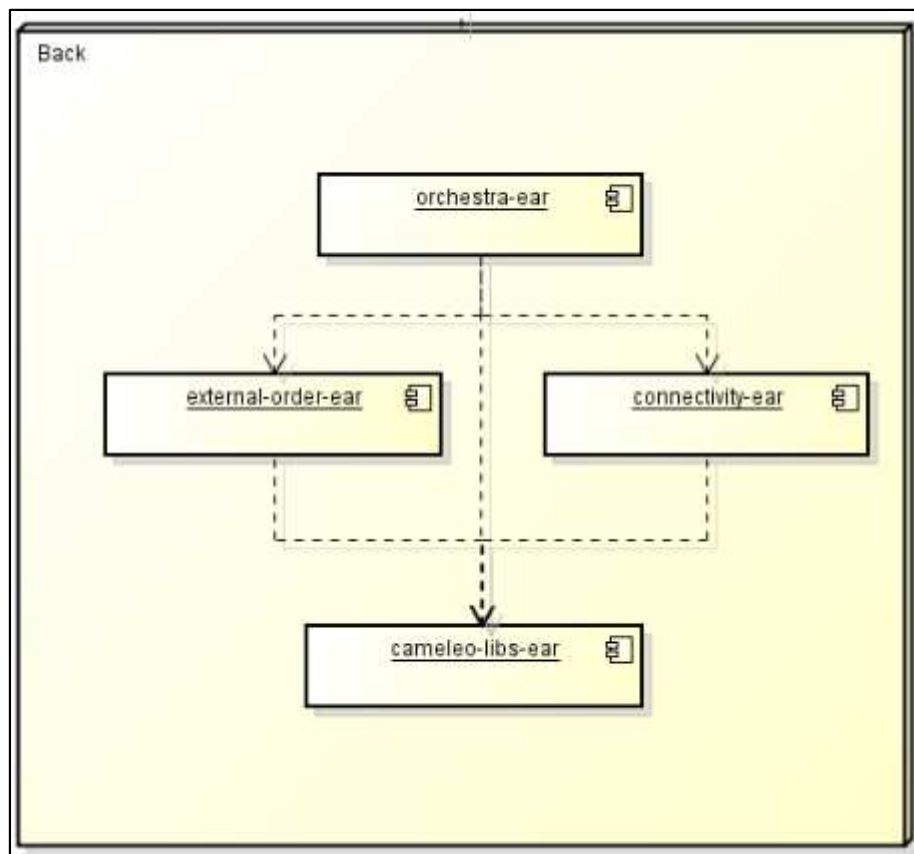


Figure 36: Deployment diagram for the back office server

On the back office we need to some projects that are also on the front, as *connectivity*, *external-order* and *cameleo-lbs*. The connectivity is needed because logically there are a lot of thing to configure on the back offices for different tour operators. The *external-order* for example will be used to connect the *bookingprocess* project of the front with the back office and on *cameleo lbs* we have a lot of classes that are used in both sides. The last capsule is orchestra and as is normal will be the project that allows us to see the back office and do everything we want there.

Still we have another server every time we build orchestra project for a client and is the internal services server. On the figure number 37 that is on the page 61 we can see that only is needed one capsule. This project is connected to the back office and to the front office and allows us to see on the back office a lot of statistics that are calculated. We can see this like the Google analytics project but only for us.



Figure 37: Deployment diagram for the statistics server

Version 8.2

The first part of the internship consisted on the work that I did from the 10th January until the 12th April, and now I am going to explain the second part of the internship that consists more or less on the last 3 months (from 12th April to 8th July).

The 8th Mars started at Travelsoft a new working method that is explained before. On that version, I worked building a new simulator, which also is explained before and it had some steps to finish for the version 8.2 (08/03/11 – 13/04/11). Finally all the mini projects about the simulator were finished on time and tested for the quality team without any problem.

Once I finished with the simulator task I started with the migrations of the static content for each client. As I explained before, there are two important folders about the static content that are the specific and generic. The generic content is the content that is created while the developing of the software and at the same time will be the same content for all the clients. On the contrary, the specific folder is a folder who has the content that the clients wants or needs for the web page. As is normal the generic content will be overwritten for the specific content.

For each version there are new modifications on the developing code and some of them have impacts on the static content. There are modifications that are only for some improvements of the software but there are others that are done because of client's requirements.

Normally the modifications that are done because any client has required something will have impacts on the specific content of that version. So, for that, each version we do the migrations of the previous version.

For example on Mars we started with the version 8.2 that we finished the 13th April. On the last days of the version I did the migrations from the version 8.0 to the 8.1 that that was finished the 8th Mars.

After the version 8.2 we started with the version 8.3 the 13th April. There I started to have more responsibilities because the tasks that I did were smaller but they had more in common with the real reservation. With this one I don't want to say that the task about the simulator hasn't got anything in common with the reservation project of the company but it's like a mini project into the reservation method. Resuming it's more individual than the other smaller tasks.

The first part of version 8.3 was to speak with my boss Mathieu, to decide with which tasks was more important to start. There are two types of tasks: the ones that are important to start because there are some others that are dependent of them and the ones that are long tasks and it's important to start with them because if all the people finish the long tasks the last days, the quality team won't have enough time to test and validate all of them. Once we decided the order I started with the development.

To explain better all the development that I did for the 8.3 version I am going to speak a little bit for each task.

Configuration of internal TOs

There are some TO that doesn't want full software that works in real time, for example because they don't have enough money to spend there or because they don't want to spend that money. So for that Travelsoft offers the possibility of creating an internal Tour Operator which saves all the data on the data base. This way we will have total freedom to manage all the data and to know in every moment the availability.

This task was created because there is a TO who asks if we could add for the internal TOs the possibility of configuring each attribute with a different way depending the TD who is doing the reservation in that TO.

There are 18 key that each TO intern can configure. I think that isn't interesting to explain all of them here, so I am going to speak about two or three of them. For example some emails are configurable like provider or annulations; there exists the possibility of configuring the number of days that a reservation could be saved without paying. Another thing which I used most to do tests was the commission.

Until this version the table was created with three columns (TO_ID, KEY and VALUE) where was possible to configure every key for each TO. The primary key was the combination between TO_ID and KEY because for the same TO wasn't possible to configure the same KEY with different values.

For this new version we have add a new column with the TD_ID and another one that has a general ID (figure 38 on the page 64) for the table and of course it will be the primary key. The ID column will be managed with a sequence that will grow one by one (figure 39 on the page 64).

```
ALTER TABLE PACKAGEBOOKING_CONFIG ADD ID NUMBER;  
update PACKAGEBOOKING_CONFIG set ID = PACKAGEBOOKING_CONFIG ID SEQ.nextval;  
ALTER TABLE PACKAGEBOOKING_CONFIG ADD DISTRIBUTOR varchar2(50) DEFAULT '';  
ALTER TABLE PACKAGEBOOKING_CONFIG ADD CONSTRAINT pk_PACKAGEBOOKING_CONFIG_ID PRIMARY KEY (ID);
```

Figure 38: Adding columns to the table

```
CREATE SEQUENCE PACKAGEBOOKING_CONFIG_ID_SEQ  
MINVALUE 1  
START with 1  
INCREMENT BY 1  
NOCACHE;
```

Figure 39: PackageBooking id sequence

Is someone has any doubt I am going to explain everything with two simple examples.

Exemple1:

Imagine that there are a lot of TDs that sell the products of the same TO and the TO wants to manage the commissions differently between all the TDs. Before doing any improvement on the project it must to use the same commission for all the TDs because on the configuration table there wasn't the possibility of configuring this part.

Now with the column distributor it's possible to earn the 25 percent of the sales from one TD and the 15 percent for the others. This is exactly the commission that will be applied on the example of the figure 40 on the page 65.

| TO | KEY | VALUE | DISTRIBUTOR |
|----|------------|-------|-------------|
| TO | COMMISSION | 15 | - |
| TO | COMMISSION | 25 | D1 |

Figure 40: Commision example 1

Exemple2:

Imagine that we want to configure a key only for one specific distributor. For that will be necessary to define the key that we want to configure only for that distributor.

On the figure 41 on the page 66 we can see that the key DISABLE_REMOTE_LOOKUP will be true only for the D2 distributor. For all the others, that key won't be defined and to finish with the table, a commission of 15 will

be used for all the distributors, because the place where we have to define the distributor is empty.

| TO | KEY | VALUE | DISTRIBUTOR |
|----|-----------------------|-------|-------------|
| TO | COMMISSION | 15 | - |
| TO | DISABLE_REMOTE_LOOKUP | true | D2 |

Figure 41: Commision example 2

Improve the management of too long fields

Until this version, we had some problems with some data base fields because we were trying to save some data with bigger size than the allowed size. There were some that of them that weren't correctly managed before arriving to the place where were saved on the data base. These problems had different impacts on the reservation software. We could have SQL exceptions which only have repercussions on the time to display, so the gravity wasn't too high. Moreover we had a problem with a specific field that didn't allow saving a dossier on the back office while the client arrives to the confirmation page. So as we can see in those examples if we don't manage correctly all the fields before storing the data on the data base, we can have really serious problems.

This task wasn't difficult but maybe one of the more important parts of the job, because if we don't save automatically the dossiers on the back office it's like if we didn't sell nothing.

To fix this part of the software we used a function called truncate as we can see on the figure 42 on the page 67. The function is called with two attributes and returns another one. Normally the two attributes that we use on the function are the

maximum size of the string and the string to truncate. The returned attribute will be the truncated string if necessary.

```
/**
 * Truncates input string.
 * @param desc the string to trunc.
 * @param max length
 * @return the truncated string.
 */
private String truncate(String desc, int max) {
    String result = desc;
    if (desc != null && desc.length() > max - 1) {
        result = desc.substring(0, max - 1);
    }
    return result;
}
```

Figure 42: Truncate function

With this method we will be totally sure that once we have truncate the string to save on the data base, will be perfectly stored.

Learned:

After this task as with another one too I knew that is very important to spend some time without touching the keyboard to write in a paper on in your mind all the error cases that we can imagine. With only five minutes of our time we can improve a lot the code to develop and of course the quality of the software.

Send and print rate details

Looking all the tasks that I did for this version, this one was the more complex, because I had to change some things on the configuration. At the same way, it was very interesting to understand where and why I had to add the new attributes on the configuration.

The subject of the task was to send a PDF file with the rate details when we do a pre-book. The good news is that the file to send it's already done so the only thing that we have to add is an attribute on the configuration to tell if we want to send the file or not and of course change some part of the code.

We read the subject and we think about that, looking for better improvements. Finally we thought about adding the possibility of configuring the possibility of sending the rate details file for all the email types. The problem is that there are configured a lot of different emails (around 20) and to decide in which email type was better to attach the file wasn't an easy decision. We continued thinking about this, and finally we decided to add the possibility of sending the file to all the different email types. This was decided because normally not all the clients want the same, so to have all the possibilities we add an attribute on the configuration called `attachQuoteFile`.

So for this, on the global configuration of the reservation project, on the place where are defined all the emails we add the attribute as on the figure 43 of the 69th page.

```
<applicationBehaviour>  
  <processusConfig>  
    <emails>  
      <emailContact attachQuoteFile="true" ... />  
    </emails>  
  </processusConfig>  
</applicationBehaviour>
```

Figure 43: Configuration for attach the quote file

After thinking a little bit more about the file that we send, we thought that maybe was interesting to add another function, but before explaining the functionality I am going to explain something about that.

On the inscription page (the second page of the reservation) there is one possibility of configuring a button to print the quotation file, send it by email or do both things. For that there is a place on the configuration where we decide if the print button is going to be printed and the functionality that it is going to do (print, send email or both).

Another thing that I think is interesting to understand the second functionality of the task is the different modes of reservation that we have on the reservation project:

- FIRM: when a client do the payment.
- OPTION: when a client reserve the travel for some days.
- PRE_BOOKED: when a client do a pre reservation.

At first on JIRA, about the task, was written that only was needed to send the rate details on a PRE_BOOKED reservation as we have explain before, but after discuss a little bit we decided that once we have the quotation file created, maybe was interesting to add the possibility of printing this file on the confirmation page depending the reservation mode that we are going to use. This one was decided because the print button that we can find on the confirmation step of the reservation project is only a screen print.

So for that, we added on the reservationMode an attribute called printQuotation as on the figure 44 at the 70th page.

```
<applicationBehaviour>  
  <processusConfig>  
    <reservationModes>  
      <reservationMode printQuotation="true" allowed="true" type="FIRM">  
    </reservationModes>  
  </processusConfig>  
</applicationBehaviour>
```

Figure 44: Configuration for print quotation

It's a Boolean and exist the possibility of configuring the attribute or not. If the attribute is configured to true, when the client does any reservation of one of our three types, the functionality of the print button on the confirmation page will be changed, now the printed page will be the quotation file with some different information like the client reference or some information of the payment, depending the reservation mode. Of course if the attribute is false will be the same that if we don't define the attribute; the print button will show a screen print, like before.

Learned:

This task didn't teach me anything that had something in common with the code but speaking with my tutor Mathieu, he taught me that if some client asks something, sometimes it's more interesting to go farther. This means that if a client asks X maybe it's interesting to develop X + Y because it could be interesting to offer to other clients and of course if the second task is related with the first we will spend less than the half of the time that we will use in the future. This is because when you work in something you can remember or know how to resolve any issue easier.

Simulator evolution

As you can read on the first part of the report, my first big job was to create a simulator to improve the usability of the testResa.jsp page and at the same time to exploit the time that was lost before when we were doing some tests.

After the first version of the simulator some of the people of the company tested it and they expressed their opinions about how to improve the simulator. There was a really big list of things to do and from them we decided to do three that were almost easy. The changes were on the xml files or on the JSP but not on the java code, so it wasn't so difficult.

The more important ideas to improve the simulator were the opinions that came from Elodie. Elodie is the product owner and she is the responsible of taking new clients for the company. She has a lot of reunions with different clients and with the new software that allows simulating the rate details page with the configuration that each clients wants, it will be easier for her to explain everything.

The three general ideas were:

1. Have the possibility to select the rate details page without all the information about the availability, prices, ...

We have two types of products: the products which we have enough information about the availability and prices (FULL) and on the other side, products which we have all the information about how about is the travel but the availability and prices and not secured (WEAK). It means that maybe we have information about them but could be wrong.

If the product that we have chosen is a FULL product, each time that we make a change on the rate details page, such as category, pension, services, etc. we won't need to make a request to the TO to know the real availability of those element because all the information that we need will be on the informationData element.

So, for doing this change on the simulator, I added FULL_PRICING option by default on the informationData template that I take for all the examples. After that on the JSP page, I added a new checkbox with the possibility to choose a request required and if this is checked I will change the FULL_PRINCING to WEAK_PRINCING.

Once I have done this part and someone choose it, when he change the category mode or options on the rate details page, all the prices will disappear and a link with the request required will appear.

2. Add the possibility to select on the compulsory and optional services if we want a quantity of options or select the number of options depending the gender (adult, child or baby).

To add this option to the simulator, first I added radio buttons to all the services where we need. The radio buttons were "quantity" or "people".

Once I know which part has chosen the user, I did java function which takes all the information about the option and changes the places where the quantity or people are defined.

This one was the most difficult part because of the option structure I add to change in two or three places and is not the same to have a normal option with a quantity or a quantity by person. Here I spend more time that the estimated because was more difficult that we thought at first.

3. Have different type of transport:

3.1. Different transportation

3.2. More flights to select on a normal departure or in a combined transport.

3.3. Improve the transport case with internal transports.

This part was easy, I took different examples on the internal catalogues and when I found some interesting examples I added to my xml file collection.

Maybe you can think that for the other examples like the quantity on the options will be interesting to duplicate the files and it was the first idea but the problem is that if we have too much files on our repertory finally the application will be too slow. To “fix” this possible future problem we decided to do a function than to duplicate the option files.

Learned:

For this task I will be brief and resuming I can tell that the client is the person who pay. This means that it's important to do well the projects that will help you on the meetings with the clients. If on a first meeting the client has good impression about the software that you want to sell, more than the half is done. For that I worked in all the versions of the sales process on the simulator.

OGONE, evolution of the algorithm to calculate the security key

Ogone is a payment method that is used for some of our clients. The purpose of the task is to update the Ogone payment method that is in our code for the new version that Ogone has launched the last month. This is important to do because there is one client who asked us if it was possible to use the last version of Ogone in their application.

The most important part is to know that for this payment method is necessary a signature that Travelsoft and Ogone will know. I think is intersection to know the differences between the old method and the new one and I am going to explain both to compare them and to understand the differences between them. The general function of the method is the next one:

- Old version
 1. Travelsoft takes 6 attributes to send to ogone and add in a string consecutively with the signature that both sides know like the last part of the string.
 2. Encode the string with SHA1 algorithm.
 3. Fill in a form with all the attributes to send, including the encoded string.
 4. Then Ogone will receive the form and with the same signature will calculate the encoded string like Travelsoft did before.
 5. Ogone will compare the encoded string that have received from Travelsoft with the string that know has encoded and if both are equals the payment will be made.
- New version
 1. Travelsoft concatenates all the attributes that is going to send to Ogone with the signature separating all the attributes.

2. Encode the string with SHA1 algorithm.
3. The same as on the old.
4. The same as on the old but to build the encoded string, it will take all the received attributes.
5. The same as on the old.

On the first version only the attributes order ID, amount to pay, currency, pspid (user id), alias (some information if we want to add) and operation were encoded with the signature on the last position of the string to send to Ogone. For the last version to secure more the algorithm, is not enough with the attributes that are encoded on the first one. Now is compulsory to encode all the attributes that are sent on the form and between each attribute we have to add the signature. So now the string to encode will be really bigger because we add all the attributes and the signature as much times as attributes we have.

The purpose of the method is to know if someone changes the data that are sent on the form and if someone has changed something cancel the payment.

Example:

Imagine that I want to use Ogone payment method to pay a travel. So for that I am going to fill in different values to send them. For example I am going to make a 2000€ payment.

If I am going to pay 2000€ to Ogone I will encode the string that after Ogone will verify if the received data is ok with the integer 2000. After filling in all the form I will send it. Now everyone can read the data that is sent and if a hacker that changes 20000€ instead of 2000€, 20000€ will be the quantity that Ogone will receive on the form.

To finish the verification, Ogone is going to encode the string with the integer 20000 and the encoded result is not going to be the same as with 2000. So after this part and when it compares both encoded strings, it will know that someone has changed any data and consequently will cancel the payment.

While I was doing this task, it happened one thing I found interesting to explain on the report. As you know Ogone takes some attributes to encode and to verify that the client and the quantity are ok. There are around 4 attributes about the client, like the address, name, surname, etc. Those values are the values that the client can enter on the inscription page.

Once I finished developing the code I made some tests to verify that with all type of characters the payment will be successful. After this, my surprise was that not with all the characters that I entered on the address of the inscription page the payment was successful. And which was the problematic character? Double quotes!

After doing some different test I thought that the problem was in Ogone because the results that I had on my side were normal but there was another part on my brain telling me that wasn't possible that a company who earns money making payments had an error like this.

Of course before sending any email to them, I concentrated more in my side and finally I found the problem. The problem came after some ending. In our part we read all the attributes to send and we encode the string to send with the whole values. The problem was that when we post to Ogone the result, was sent by an URL. On the URL after the path we add all the attributes with their values. We couldn't use the URL encoder here because if we had done like this, maybe some data will be changed. As we send the data like an attributes on the URL the problem with the double quotes was that Ogone cut the string too soon.

Double quote error example:

Imagine that we have two attributes to send to Ogone:

- *ADDRESS: rue Ampère "A510*
- *QUANTITY: 1000*

If our signature is "signature1", the string to encode will be:

- *ADDRESS= rue Ampère"A510signature1QUANTITY=1000signature1*

If we pass through the encoding algorithm this string we will have one result, like for example: "encoded1" (of course the results are not like this, but this is only one example).

The post that we will send without an URL encoder will be the next one:

- *www.testogone.com?ADDRESS="rue Ampère"
A510"&QUANTITY="1000"&ENCODED="encoded1"*

So once that Ogone receive the post, it will read the attributes from there and it will save this data:

- *ADDRESS: rue Ampère*
- *QUANTITY:1000*
- *ENCODED=encoded1*

Ogone doesn't know, as its normal that all the address is until the next attribute because it finds the first double quote before the last part of the address (A510). So, once with the attributes and with the same signature (is known for both sides) it try to encode the string with the algorithm but they it doesn't try to encode the same string. They will try to encode this one:

- *ADDRESS= rue Ampèresignature1QUANTITY=1000signature1*

Of course the result of the encoding won't be the same than ours, because the string to encode is not the same. So when Ogone compare his encoded string, imagine that is "encoded2" with the string that we sent them on the post "encoded1" it will see that they aren't the same and we will answer us with an error.

After this example I want to remark too that with this I found a bug that already was on the reservation of Travelsoft on precedent versions. To fix the problem we changed the regular expression that limits all the strings that we can find on the inscription page.

I know that isn't normal to have a name with double quotes or an address, to be hones I doubt that doesn't exist anything like that. The only problem that we had was that we couldn't make the payment, and if any client really wants to buy a travel, he is going to use real values. Even so, I know that is important to don't let to the client to enter that type of values on a textbox that after can give use problems. In any case this wasn't a problematic case but we must pay close attention to everything that is done especially if the client participates.

To finish this small part, I want to stress that it's very important to think in all the cases that exists all around the computer engineering. So I think that **it's more important to spend a little more time estimating all the cases and trying to find all the problems that we can have**. Like this we would have really less issues.

Sequence diagram:

I was going to add a sequence diagram for the part that I developed but as I did the activity diagram there I found interesting to draw this diagram for other functionality.

As we can see on the figure 45 on the page 80 we have four parties involved on the diagram. The first one will be the client who wants to pay a trip on the booking process. This one will show to the client the page where he can pay. After this the client will send the payment to Ogone that verifies the data and send the response to the booking process. Then if the payment is refused the booking process sends to Ogone a redirection to an error page that Ogone will send to the client. If on the contrary the payment is accepted at the same time two messages will be sent. The first one is a redirection to Ogone that Ogone sends to the client to a waiting page and the second one is a book request to the tour operator. With this one the client will be waiting on the waiting page until a new redirection page is received from the booking process. The answer will be different depending if the book was possible or not. So if the book was possible everything goes correctly but if the book couldn't be possible we have a problem because the client has pay but he hasn't got the trip. So then the booking process sends to Ogone a cancel request and at the same time an error page to the client.

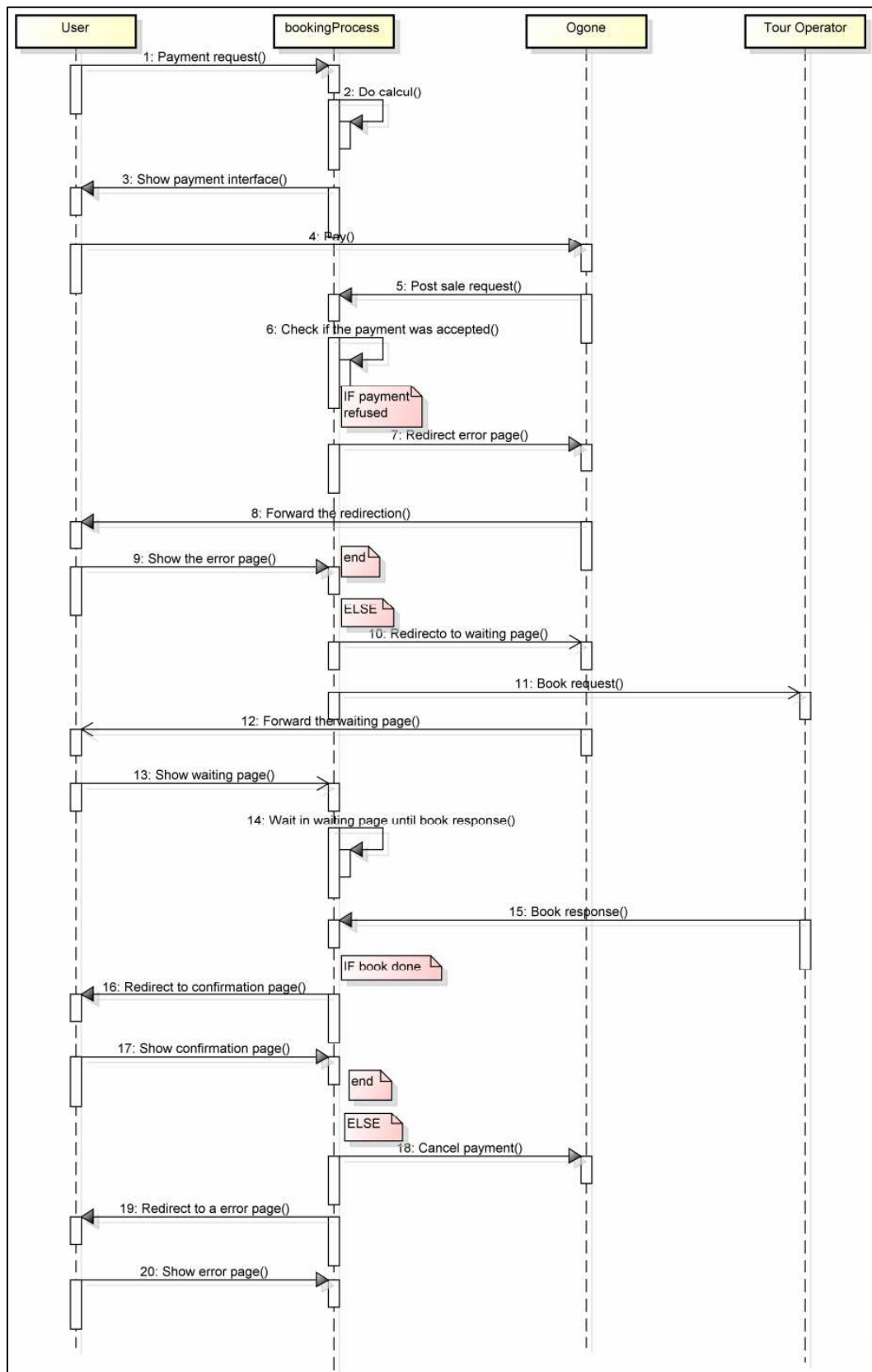


Figure 45: Sequence diagram for Ogone payment

Learned:

As I explained before in this task before add something in production and to prevent any problem it's interesting to test all the different cases that can pass around your brains.

Client Migrations (8.1 → 8.3)

On all the versions there is a part that is the migrations. There isn't really a real developer work but it is so important for us to do because we are the people who before did the changes.

As I explain on the first part of the report, on the sales process of the project that Travelsoft can offer we have for each client on the ftp a folder with the static content. There is a generic folder and a static one. The generic folder is the static content that Travelsoft has to make working the entire reservation project. In the other side we can find the specific folder and is the folder that each client configures as they want to have the design of the pages different as the Travelsoft one. For example on the velocity files it's possible to configure the different blocs that we want to show on the internet and also the order. There exists the possibility of changing the css files to change all the colors of the webpage and configure with the requirements of each client.

When we develop something on the java code we try to don't change the static code if isn't compulsory, because after if that client has the file that we have changed on his specific folder and our change has impacts on that file, we must modify the part of the code on the specific file too.

Example:

I am going to explain a case that happened to me on this version. There is a quotation file to send in any email as I explained on the task TPCR-7367.

As we have 3 different payment methods and as we don't want the same title message depending the reservation mode we were forced to change the titles and messages on the summary properties file.

Before all the development the title DEVIS was showed in all the quotation files, regardless the reservation mode. But now we want to show these titles:

- To send mail on the inscription page: Devis
- For all pre booked reservations: Devis
- For all reservations without payment: Option
- For all paid reservations: Reservation

This means that on the developing task we have to change the velocity file that is the file that builds the file that is going to be sent and also the properties file to add the new messages with the new keys.

The message that he had for the titles of the quotation file regardless the reservation mode that we have used was: *reservation.quotation.devis*. Of course that with only this message is not possible configure for the three others so we decided to change this one and add three more and finally the messages were these ones:

- *reservation.quotation.devis.title*: Devis
- *reservation.quotation.PRE_BOOKED.title*: Devis
- *reservation.quotation.OPTION.title*: Option
- *reservation.quotation.FIRM.title*: Reservation

After that we changed the part where on the place which was added directly the `reservation.quotation.devis` on the `quotation.vm` file for a small code part to compare the reservation mode and add the corresponded title.

The problem of this is that if any client had the title on their specific file or on the `quotation.vm` file configured with that message we were forced to change one by one on the files of each clients and yes, it happened.

The good thing is that wasn't difficult to change this but yes for more than the half of the clients and that's time to update the directory of the client, search the file to change, change the file, commit the changes,... As I told on the report is easy to do but if we have to do for a lot of projects maybe is better to think twice the modification that we are going to do on the code.

There was another possibility, and was to add the three new properties and to don't change the one that already exists. This could be right if we look on the migrations but if we want to have a correct structure on the properties file was important to add for all of them the part `".title"`. This was added because reading only the properties key will be interesting to know where that key is or why we created that key. So because of this, I decided to change the key. I prefer to have more understandable the properties file in exchange for working more on the migrations.

Learned:

Here there is not so much to learn because it's to update the client files if it's necessary but of course I can stress here that this task is very important to have all the projects of every client ready to update them to the current version.

Version 8.4

I wasn't going to explain on the report nothing else from the version 8.3 because is the version where I am working now but I found interesting one task that I did because it change something about the static content that I explained before.

Separation of the generic content and specific content

Until now for each client we saved on the ftp with the static content and the specific content. Of course exists the possibility of changing this and improving too but the problem is the number of impacts that we can have depending the type of changes.

To understand perfectly what and why we did with the static content I am going to resume fast what had before the static dossier of each client:

- csvlocal: content specific for each client (properties and velocity).
csvweb: content specific for each client (css, pdf, image, etc.).
- TS: saved and order folders. On the first we save files from the simulator and on the second we save files every time we do a reservation, to control that everything was well done.
- CS: some files about Europ insurance.
- Tsvlocal: generic content about the version that the client is using (properties, velocity and some configurations).
- tsvweb: generic content about the version that the client is using (css, pdf, image, etc.).

It's important to tell that for each static content (specific or generic) we have two different folders that are web and local. Web folder will be charged with apache and local folder with Jboss, and because of that we divide both.

On this version we decided only to change the place where we saved the static content of each version. It means that each client wouldn't have the tsvlocal and tsvweb folders with the static content of each version.

Now each client is going to have in their folder on the ftp only specific information about them. With this methodology we are going to win a lot of storage size on the server. Before this we had as much static content folders as clients but now we will have as much folders as different versions are used by the clients that will be considerably less.

We use the name tsv for the generic folders and csv for the specific ones. So after all the change we are going to have on the tsvlocal folder the specific configuration for each client. I know that isn't so normal to save this configuration file there but we decided to don't move now because behind that file there are a lot of permissions and would bring a lot of work to make everything work correctly. Apart the sales process configuration file we will save on this folder some more configurations (depending the TD) for the same reason.

Before start developing the improvement we decided that was interesting not change dramatically the static content structure and we save the old configuration. It means that we will change the repositories of the clients when we want and we can continue using the old structure.

On the configuration we had an attribute called staticFilePath that tell us where we have to find the static content. Now based on that attribute we only have added one more called specificStaticFilePath as we can see on the figure number 46 that is on the 86th page.

```
<bookingProcessConfig>
  <globalConfig>
    <technicalInformation
      staticFilePath="/var/data/static/generic/8.3/front"
      specificStaticFilePath="/var/data/static/Carrefour/front">
    </technicalInformation>
  </globalConfig>
</bookingProcessConfig>
```

Figure 46 : Configuration for static content

How the static content folders work is simple. If we have declared only the `staticFilePath` on the configuration we will work like before, one folder with all the necessary content for each client. But on the case of having also the `specificStaticFilePath` declared we will find on that folder all the specific content about the client and on the `staticFilePath` we will find the generic content of the version that the respective client is using on the sales process.

Example:

Imagine we have 40 clients using five different versions. Normally between one version and other one there aren't so many changes on the size of the files. So we can estimate that the size of each version is almost the same.

This is the simulation about the used version for the clients:

- V6.2: 3 clients.
- V7.1: 5 clients.
- V8.0: 13 clients.
- V8.1: 10 clients.
- V8.2: 9 clients.

As we have 40 clients with the old version we were saving 40 times the size of the static content but now we only are saving five times that size. This means that we are **improving eight times** the efficiency about the storage.

Learned:

On this task I learnt that is very important to think in possible improvements before than could be late. This means that for us, if we change now for all the clients we will have work changing the configuration files of all of them but will be less than in some years because if the things go well there will be much more clients.

Apart of that and of course more important is the performance. Linking this problem with the other, we can tell that if we grow up like a company and if in a few years we had not changed this we would have performance problems because we would be saving on the server a lot of duplicated information.

Also another time I realized that all the changes that seem easy won't be. For example in this case was logic to change the configuration files from the tsvlocal folder to the CS or TS folders but once we saw all the impacts about the permissions we decided don't to do this.

Conclusions

During my stay at Travelsoft, I have learnt so much technically as a development engineer. As I explained before, my first days at the company were devoted to training. I had more than one meeting with different people who thought me a lot of interesting thing. Also there is a big intern library where I read a lot of things that after I have used all the days. Then, very quickly the learning time converted in developing time. In this second time is the time where I have learnt more, because I worked in a team that helped me in everything I needed. Gradually I became familiar with the new technical environment at the expense of performing more tasks. It is the knowledge that I acquired at the university which allowed me to adapt into the worker live.

Project management it is the work in the future I would like to do. I know that now it is impossible to start being a project manager because I am a beginner but really it is the part that catches my attention. So far I had not the opportunity to see how works a project manager although at the university I learnt the theory. In my days at Travelsoft, I worked alongside a project manager and little by little I learnt something about it. The most important thing to be a project manager is to work before like a normal developer and when you know a lot about the tool or the project where are you going to work. With this one you will acquire the enough knowledge that combined with how to manage a team will let you leading a team correctly.

One of the things I would like to stress is the opportunity that Travelsoft gave me to work in a team. Without this part, I had not learned even half of the part that I finally learnt. The welcome that I have on the team was almost immediate. The atmosphere was perfect. This allows you to work calmly and finally you will give more to the company. Out of my development team the environment was fantastic. I joked a many days with the director of the company about the football. Also I made jokes

with another project manager about the handball and like those I could tell a lot of histories. Resuming the people in Travelsoft is a small family with whom you can count on whenever you need it.

Another think that is going to make me rise like an engineer is the new technology which I am working now. Struts2 is the ideal framework to build web pages with Java. I have worked before a little bit with the first version but not so much and as all the work that I did at Travelsoft was related with this framework I learned a lot about this modern tool.

I could be writing here a lot of things, but if I have to choose another one between all the groups of things that I have learned is how to work with an agile software development. It was the first time I have worked this way and it made me see the things differently. Now I know the importance that has each person on the duration of each iteration.

Future lines

On this part of the report we could add a lot of things because the reservation mode of travelsoft is really huge. But if I have to choose something I will remark the simulator that I did and also big works that will be good the reservation mode.

Starting with the simulator we can tell that if we arrive to the rate details page configuring all the different blocs is not possible to go on, on the reservation. This is because once we have decided the different blocs we want for the travel, a quote request is done to the tour operator to verify if the prix and the date still are available. So a good improvement for this software will be, to have the possibility of going until the confirmation page step by step.

To do this I have thought two methods. The first is to develop a function that will replace the real actions that will allow us to simulate a whole reservation. To do this, the information data template must be complex, because we need a lot of information. The second way is create a product that has all type of combination that we can reach. So like this we could have a real reservation but this part is harder than the first way and won't be cool to spend too much time in a thing that maybe we won't use a lot.

Another thing that will be interesting to improve on the simulator is to select the number of the people for the travel. Like this all the rate details web page will be appropriate. I am telling this because for example there are some cases for the lodging that are configured only for a pair, other ones for 3 people and there are two cases configured for two pairs. Like this we can show the entire page correctly but will be much better if we could select the number before. Also the services will be displayed equivalently with the number of people going to sleep in the hotel.

Speaking more in general about the reservation method there are some parts that I found strange something that I think that is important weren't developed, the shopping cart. Even a normal online shop has this one developed, so one day speaking with my boss I asked him if we have this possibility on the catalog or not and he told me that no but it was oncoming because a new client asked about this. So I think that the most work will be on the side of the catalog to save the information of all the travels but for us will be difficult because we will need to multiply the times of the reservation info objects.

As a travel company we are, I think that is interesting to have the possibility of selling only hotels. This possibility is going to start fast because a new Spanish client (Viajes Carrefour) has asked. In France normally all the stays are fix in dates and you can select or not but isn't normal to book only the hotel so as I have explained before this will be fantastic to open the horizons.

Selling only transports could be positive too. Always this type of improvements brings new clients and to have more clients is the goal to make money. Until now isn't possible to buy a transport without the lodging.

It will be possible to add more and more improvements because the dimensions of the software but I have explained only the improvements that for me can change a lot the reservation method, and of course improvements for the software that I create.

About the smaller advancement will be to have the possibility of having checked some optional service by default. It means for example that through some received id we will be able to check by default unless it is an optional service. This is done for example on the WebPages to buy plane tickets with the insurances and if you don't pay attention some charges are added to the total amount.

Another think that I think that will be interesting to do is the way which we can book a travel that is on request. I know that showing this like a business it's normal but if you think like a client is not too lawful. For example if we choose a stay for which we don't have availability confirmed we can continue the reservation and finish a reservation that finally will be on request. This is displayed in a string on the last part of each page. For me could be fair if the client has any pop-up telling that the product is on request or also offer departure dates near the date that the client has chosen.

References

- [1] J2EE: <http://download.oracle.com/javaee/1.4/tutorial/doc/>
- [2] SCRUM: <http://www.scrum.org>
- [3] Open source software: <http://www.opensource.org/osd.html>
- [4] Eclipse Environment: <http://www.eclipse.org/>
- [5] Subversion: <http://subversion.tigris.org/>
- [6] Maven: <http://maven.apache.org/>
- [7] Bamboo: <http://www.atlassian.com/software/bamboo/>
- [8] Jenkins: <http://jenkins-ci.org/>
- [9] Jira: <http://www.atlassian.com/software/jira/>
- [10] JAXB: <http://jaxb.java.net/>

Prácticas en Travelsoft

Desarrollo Orientado a Objetos en una aplicación J2EE
de Gestión y Reservación de viajes

Mikel Otxandorena

Indice

- Presentación empresa
- Mi trabajo en Travelsoft
- Conclusiones
- Demo

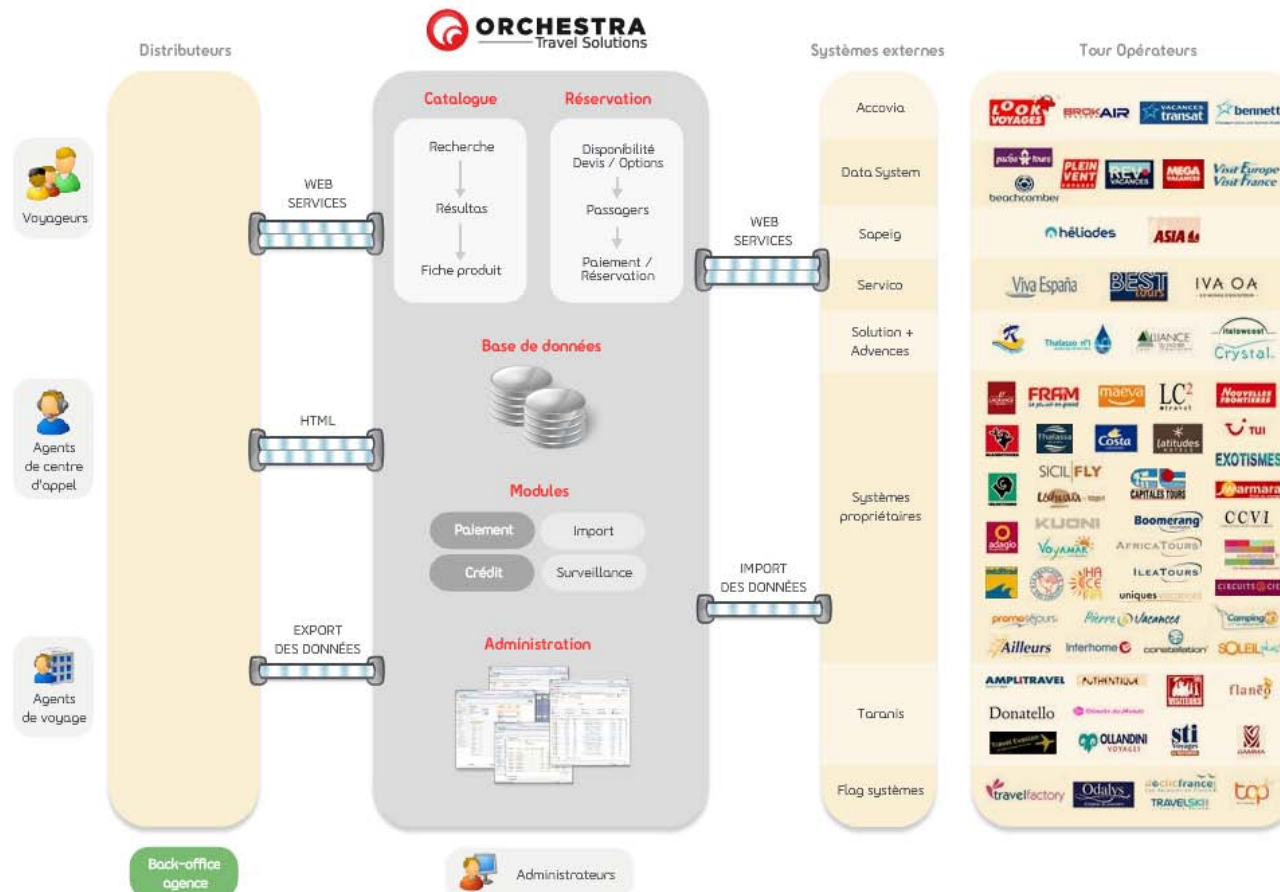
Presentación empresa

- Travelsoft, empresa dedicada al negocio de los viajes
- Creada en 2000 por Marko Vujasinovic, Christian Sabbagh y Pablo Carrington
- Situada en Avenida de la Opera 38, 75002 PARIS
- 60 trabajadores

¿Qué trabajo realizan?(1/2)

- Plataforma de distribución Cameleo:
 - Importar productos desde los Tour Operadores (TO)
 - Distribuir los productos a los diferentes Tour Distribuidores (TD)
- Añadir prestaciones externas al viaje(Seguros,...)

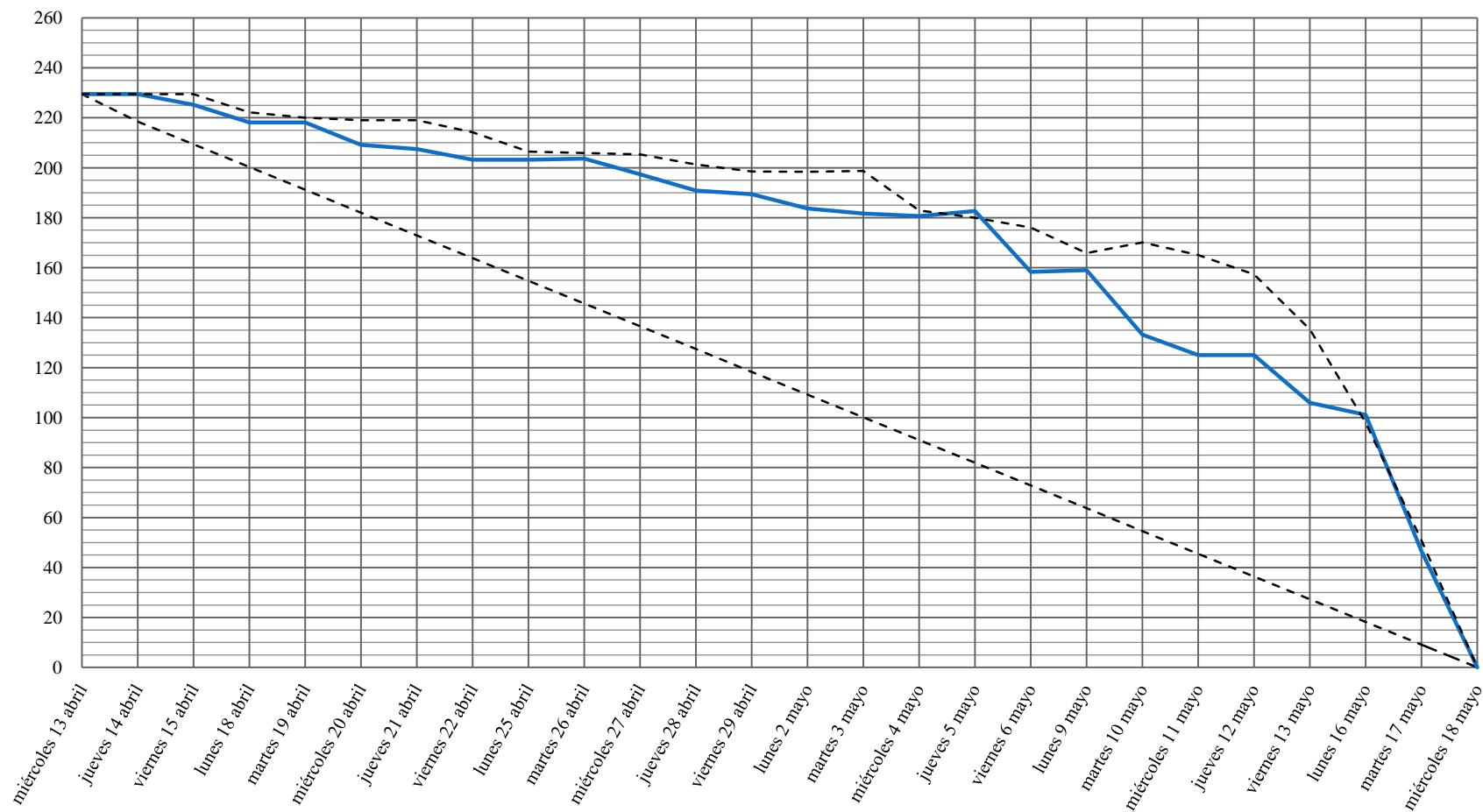
¿Qué trabajo realizan?(2/2)



Metodología de trabajo(1/2)

- Todo el entorno esta desarrollado mediante una metodología ágil
 - Utilización de una pizarra
 - 3 reuniones por semana
 - Test de calidad a lo largo de toda la versión
 - Mini demo para cada tarea

Metodología de trabajo(2/2)



Mi trabajo en Travelsoft (1/3)

- Modulo de reserva
 - Problemas:
 - Productos inexistente
 - Respuesta del TO muy larga
 - Soluciones:
 - Desarrollo de un nuevo software
 - Utilizable por muchas personas(diseñadores, responsable de producto, desarrolladores, ...)
 - Facilitar la manera de realizar los test

Mi trabajo en Travelsoft(2/3)

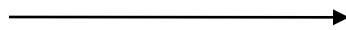
- Instrumentos utilizados
 - J2EE: Java Enterprise Edition
 - JAXB data binding: trabajo con ficheros XML.
 - Velocity: plantillas para la estructura de las páginas
 - JavaScript: funcionalidades de las páginas
 - Properties: guardar mensajes del contenido estático
 - Struts 2: framework utilizado.
 - Jira: almacenamiento de tareas
 - Bamboo y Jenkins: integración continua

Mi trabajo en Travelsoft(3/3)

- Bloque debug
 - Ver la configuración y objeto principal (XML)
 - Controlar el perfil utilizado
 - Registrar ficheros
- Generador de plantilla con características del viaje
- Modulo de reserva desde fichero XML
- Envío de mail desde fichero XML

Bloque debug

testResa.jsp



Características viaje



Inscripción



Pago



Confirmación

DEBUG TOOLS

| Profil data information | Received profile | Canonical profile | Generate XML page |
|--|---|--|--|
| <ul style="list-style-type: none">• Information data• Configuration• Clean cache | <ul style="list-style-type: none">• defaultResConfig:• channel: RESFORM• organization: org1• user: user1 | <ul style="list-style-type: none">• defaultResConfig:• channel: RESFORM | File prefix: <input type="text"/> <input type="button" value="Save"/> |

bookingprocess : 'prerecette-front' instance on: 'baltimore' - 8.2-TEST - Built on 20110324-1143 by team.

Generador de plantilla(1/2)

- Carga rápida de la página de características del viaje; sin realizar llamada al TO
- Seleccionar los elementos del viaje que queramos
- Rellenar la página mediante información que hay guardada en diversos ficheros XML

Generador de plantilla(2/2)

Package

☒ Lodging

☒ Simple ☐ Multi

☐ With Options
☒ With Pension
☒ With Category

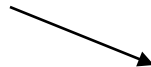
☒ Transport

☒ Simple ☐ Combination

☐ Pre
☐ Post
☐ Intern
☒ Normal

☐ Taxes

☐ details



Veuillez confirmer ou modifier votre choix

Forfait principal

HÉBERGEMENTS ET TRANSPORT

Hébergement

Product name from testResa

Période : du 10 juillet 2011 au 17 juillet 2011

☒ 1 x Chambre double (2 adultes) [masquer](#) Disponible Inklus

Prestations

☒ Pension DEMI-PENSION : Inclus

Chambre 1

☒ Catégorie Chambre double BAIN, minimum : 1 adulte age maxi enfant=12 : Inclus

☐ 2 x Chambre simple (1 adulte) [afficher](#)

Transport

PARIS ORLY OUEST, dim. 10 juil. >> Tenerife (Aéroport Reina Sofia), dim. 10 juil.

Aller

PARIS ORLY OUEST dim. 10 juil. à 20h10 → Tenerife (Aéroport Reina Sofia) dim. 10 juil. à 23h30 Disponible

Retour

Tenerife (Aéroport Reina Sofia), dim. 17 juil. >> PARIS ORLY OUEST, dim. 17 juil.

Tenerife (Aéroport Reina Sofia) dim. 17 juil. à 14h20 → PARIS ORLY OUEST dim. 17 juil. à 19h15 Disponible

Reserva desde fichero XML

- Ir a las diferentes etapas de la reserva desde fichero XML
- Simulación rápida

Test template file

| | |
|---|--------------------------------------|
| Server path file | Selected step |
| <input type="text" value="/var/data/tmp/CONFIRMATION.xml"/> | <input type="text" value="Default"/> |
| <input checked="" type="radio"/> Configuration by default <input type="radio"/> Configuration from file | |

Bookingprocess : 8.2-TEST built on 20110321-1635 by Mikel.

Envío de mail desde fichero XML

- Simulación de todo tipo de mails

Send template mail

| | | |
|--|-------------------|-------------|
| Server path file C:/ts/data/workspace/bookingprocess_GENERIC | Motive Default | Type All |
| Use custom mail (disable config mail) mikel.otxandorena@travelsoft.fr | | |

Submit

ngprocess : 8.2-TEST built on 20110324-1123 by Mikel.

Client tracking

Business tracking

Client pre booking confirmation

Business pre booking confirmation

Insurance Subscription Error

Book Price Alert

Quotation

Conclusiones(1/2)

- Importante aprender el software antes de empezar a desarrollar
- Escribir en un papel o pensar la solución antes de tocar el teclado
- Puede ser interesante ir mas lejos de los requerimientos del cliente
- Buen rendimiento a lo largo de cada modificación del software

Conclusiones(2/2)

- El cliente es la persona que paga
- He aprendido a trabajar en equipo
- Mejorar mis conocimientos con Struts2
- Me ha gustado trabajar con una metodología ágil

Muchas gracias por vuestra atención

¿Demo y preguntas?